SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA DA UFBA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Volker Kible

SOFTWARE-DEFINED RF REFLECTION COEFFICIENT MEASUREMENT BLOCK BASED ON UNDER-SAMPLING DOWN-CONVERSION

Salvador, 2021

Universidade Federal da Bahia
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

SOFTWARE-DEFINED RF REFLECTION COEFFICIENT MEASUREMENT BLOCK BASED ON UNDER-SAMPLING DOWN-CONVERSION

Volker Kible

TESE SUBMETIDA AO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
DA UNIVERSIDADE FEDERAL DA BAHIA
COMO PARTE DOS REQUISITOS NECESSÁRIOS
PARA OBTENÇÃO DO GRAU DE
DOUTOR EM ENGENHARIA ELÉTRICA.

Área de Concentração: Processamento de Sinais
Linha de Pesquisa: Microeletrônica em RF

Orientador:
Prof. Dr. Robson Nunes de Lima – UFBA

Salvador, Bahia, Brasil
Maio de 2021

K46 Kible, Volker
Software-defined RF reflection coefficient measurement block based on under-sampling down-conversion / Volker Kible. – Salvador, 2021.
123 p.: il. color.

Orientador: Prof. Dr. Robson Nunes de Lima.

Tese (doutorado) – Universidade Federal da Bahia. Escola Politécnica, 2021.

1. Coeficiente de reflexão. 2. Refletômetro. 3. Algoritmos. I. Lima, Robson Nunes de. II. Universidade Federal da Bahia. III. Título.

CDD: 511.5

SOFTWARE-DEFINED RF REFLECTION COEFFICIENT MEASUREMENT BLOCK BASED ON UNDER-SAMPLING DOWN-CONVERSION

Volker Kible

Tese de Doutorado aprovada em 25 de junho de 2021 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Robson Nunes de Lima
Orientador e Presidente da Sessão – UFBA

Prof. Dr. Maicon Deivid Pereira
Membro Interno – UFBA

Prof. Dr. Bernardo Rego Barros de Almeida Leite
Membro Externo – UFPR

Prof. Dr. Fabrício Gerônimo Simões Silva
Membro Externo – IFBA

Prof. Dr. Paulo Márcio Moreira e Silva
Membro Externo – UNIFEI

Table of Contents

List of Figures

List of Tables

Acknowledgements

Special thanks I would like to express towards my wife Karolinne Brito Kible and my son Benjamin as well as my parents Dieter and Rose Kible for their patience and helping me to keep my spirits up.

My deepest gratitude goes to my advisor Prof. Dr. Robson Nunes de Lima for his help and persistence, also over the distance of an ocean.

Furthermore, I would like to thank Prof. Dr. André Zimmermann, André Bülau and Florian Janek from Hahn-Schickard Stuttgart for their practical support in writing scientific papers and in placing and soldering some difficult components as well as help using the institute's equipment and Mohamed Elkeshti for his contribution to chapter 7 under my supervision. I also would like to express my gratitude to the staff of the Programa de Pós-graduação em Engenharia Elétrica (PPGEE) of the UFBA for their help with concluding this work.

Finally, I would like to thank the FUNDAÇÃO DE AMPARO À PESQUISA DO ESTADO DA BAHIA (FAPESB) for their financial support during the beginning of the work.

Abstract

In general, there are two principles to determine the reflection coefficient: one is based on interferometry, as used by the six-port reflectometer. Another method is based on the separation of reflected and incident waves and calculating their ratio. This method relies on directional couplers or similar devices and is used in most vector network analyzers. Based on this principle, a simplified reflectometer for measuring the load impedance of a power amplifier, was developed. This reflectometer measures the reflection coefficient at radiofrequencies. An interesting strategy to reduce non-idealities and size of a reflectometer system is to carry out different processing steps for the software domain. The challenging in this case are the processing requirements for real-time analysis of the radiofrequency signals. To counteract the processing requirements, sampling effects in the inevitable analog-to-digital converter are exploited.

Physical and mathematical constraints for software-defined reflectometers based on under-sampling down-conversion and their design theory are defined. Then the validity of the concept and design theory were proven by computer simulations and automated measurements against known references. The reflection coefficient measurement errors on a discrete printed circuit board setup at 400 MHz were below 2% using under-sampling down-conversion. This is a particularly promising result when compared with similar systems. Under severe mismatch conditions, the under-sampling scheme failed, but these cases present a high standard deviation and are thus easily recognizable. Responsible for the failure of the under-sampling are harmonics generated at the output of the unprotected power amplifier. Various promising remedies for the under-sampling failure are outlined. However, as the system is planned to be part of an automatic impedance matching system, extreme mismatch conditions should normally not occur.

Our study shows that a software-defined reflectometer for radiofrequency (300 MHz to 3 GHz), based on standard hardware components and implementable in integrated circuit technology, for dynamic impedance matching purposes, is feasible.

Resumo

Em geral, existem dois princípios a partir dos quais é possível determinar o coeficiente de reflexão: um é baseado na interferometria, como no refletômetro de seis portas. O outro método é baseado na separação das ondas refletidas e incidentes. Este método se baseia em acopladores direcionais ou dispositivos semelhantes e é usado na maioria dos analisadores vetoriais de rede. Com base nesse princípio, foi desenvolvido um refletômetro simplificado para medir a impedância de carga de um amplificador de potência. Esse refletômetro mede o coeficiente de reflexão em radiofrequências. Uma estratégia interessante para reduzir as não-idealidades e tamanho do sistema é realizar diferentes etapas de processamento em *software*. O desafio, nesse caso, são os requisitos de processamento para análise em tempo real dos sinais de radiofrequência. Para balancear tais requisitos, os efeitos de amostragem no conversor analógico-digital podem ser utilizados.

Restrições físicas e matemáticas para refletômetros definidos por *software* com base na conversão de subamostragem e sua teoria de projeto são definidas. Em seguida, realizaram-se simulações em computador e medições automatizadas para comprovar o conceito e a teoria do projeto, comparando os resultados com referências conhecidas. Os erros de medição do coeficiente de reflexão, em uma configuração discreta a 400 MHz, foram inferiores a 2% usando subamostragem de conversão descendente. Esses resultados são promissores quando comparados com sistemas semelhantes. Em condições de forte desadaptação, o esquema de subamostragem falhou, mas esses casos apresentam um desvio padrão alto e, portanto, são facilmente reconhecíveis. Tais falhas da subamostragem são causadas pelos harmônicos gerados na saída do amplificador de potência desprotegido. Várias soluções promissoras para a falha de subamostragem são apresentadas. No entanto, como o sistema é planejado para fazer parte de um sistema automático de adaptação de impedância, condições extremas de desadaptação não devem ocorrer normalmente.

Nosso estudo mostra que um refletômetro definido por software para radiofrequência (300 MHz a 3 GHz), baseado em componentes de hardware padrão e implementado em tecnologia de circuito integrado, para fins de adaptação dinâmica de impedância, é viável.

1 Introduction

1.1 Theme and Justification

Mobile RF telecommunication systems are of central importance for the modern society. Amongst others, they are used in the fields of consumer electronics, military, medical technology, and industrial automation, and will gain even higher pervasion with current developments such as the internet of things and industry 4.0 [1].

Such systems use antennas for converting the internal transmission line bound RF signals to electromagnetic radiation traveling between devices. These antennas are often hidden inside the case of the mobile devices and thus can be in direct proximity to the users' bodies, a condition which can frequently be observed in cellular phones. It is known that the electric properties of the antenna, especially its input impedance, depend on the surrounding materials. Because these are generally unknown and often varying, the input impedance can also not be assumed as constant. The proximity of a user head or hand [2], [3] can alter the antenna input impedance significantly, possibly leading to a varying impedance mismatch between the antenna and adjacent building blocks of the digital transceiver.

Digital transceivers are the heart of modern RF telecommunication systems. They comprise of various building blocks as shown in Figure 1.1. The building blocks adjacent to the antenna are the low noise amplifier (LNA) in the reception path and the power amplifier (PA) in the transmission path. While the effect of varying antenna input impedance on the LNA is generally a reduction of input power and can be compensated in the following intermediate frequency variable gain amplifier (IFVGA), the effect on the power amplifier (PA) can be more severe.

For the PA, the varying antenna input impedance results in a changing load reflection coefficient. More exactly, the load reflection coefficient can change in phase and magnitude. Ideally, the magnitude would be zero. Increasing magnitudes of the load reflection coefficient imply that an increasing amount of the transmission power is reflected back to the PA. To bridge transmission distances of few kilometers as usual in mobile communications, the output power of the PA must be high compared to the other parts of the system (except in ultra-low transmission power devices). As a total reflection of the output power back to the PA can occur, the signal amplitude at the PA output can get very high. This can cause degradation of the PA's properties such as linearity or efficiency and can in extreme cases even damage the PA if not protected [4].

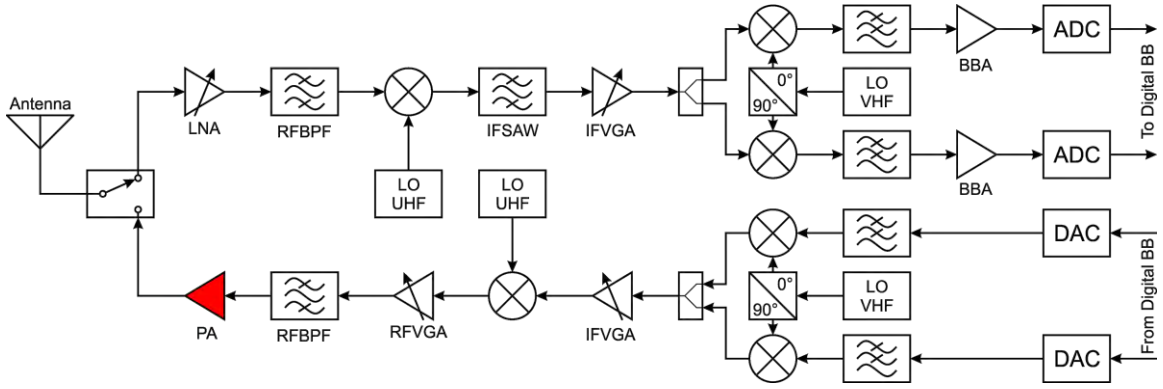

Figure 1.1: A Typical Digital Transceiver (Superheterodyne)

So, it is of crucial interest to find countermeasures against the effects of a change of antenna input impedance. There exist various approaches, from dissipating the reflected power and automatic gain control of the PA [5], out-phasing amplifiers [6], hybrids [7] and distributed active transformers [8] to variable impedance matching networks [4], [9]-[11].

Approaches other than the variable impedance matching networks have in common that they can reduce the susceptibility of the PA output to the impedance variation by some factor, but the mismatch and its effects still occur.

Variable impedance matching networks offer the possibility to dynamically match the varying antenna input impedance to the PA output impedance and thus keep the load reflection coefficient close to the ideal zero. For this operation, the current antenna input impedance must be monitored continuously. The combination of a reflection coefficient measurement block, the control block, often incorporating software, and the variable impedance matching network is called an automatic impedance matching system, as shown in Figure 1.2.

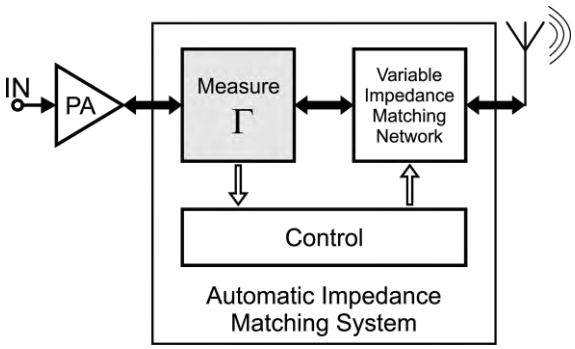

Figure 1.2: An Automatic Impedance Matching System

This work will focus on the reflection coefficient measurement block of such an automatic impedance matching system. Compared to the variable impedance matching network and the control block [12]-[13], relatively few research activities are ongoing in this field, making it interesting for contribution. Furthermore, such a measurement block can be used not only in an automatic impedance matching system, but also in instrumentation, for example for measuring the humidity content in material samples via interaction with RF waves [14]. For this purpose, only an RF signal source and an antenna

which couples the RF waves into the material sample, for example a slotted hollow wave guide [15], are needed besides the measurement block.

In [16], the use of injection locking and quadrature amplitude demodulation for measuring the load impedance was explored. Research on possibilities to correct shortcomings of this method such as offset problems, non-idealities of the mixers and others gave rise to a completely new approach: A software defined reflection coefficient measurement block based on under-sampling down-conversion.

The new approach is expected to consume less power because less circuitry is involved. The involved digital or mixed signal circuitry, such as the analog to digital converter, would be necessary in any case. Analog circuitry is replaced by computations in the digital domain with their inherent exactness. This comes at the cost of raised requirements at the analog to digital converter, especially its sample and hold circuitry. The necessary sample clock can be derived from the UHF local oscillator or from the transmit signal itself, guaranteeing a fixed frequency relationship.

By means of using a non-zero intermediate frequency in the digital domain, the effects of nonidealities of the remaining analog circuitry such as offsets are avoided. Under-sampling down-conversion is used to limit the complexity of the analog to digital converter and digital signal processing requirements, by reducing the necessary sample clock frequency (in the lower MHz range instead of UHF) and avoiding other analog blocks.

As already stated in [16], the system frequency was chosen to be in the range of 400 MHz (low UHF). This frequency is sufficiently high to get high frequency effects (it allows for extrapolation into the middle and higher UHF range), but also so low that the wavelength is big enough (in the 0.7 m range) to make it challenging to integrate wave guide structures. Also, antenna mismatch occurs often in the MICS band (Medical Implant Communication Service, 402 to 405 MHz, limited to 25 μW output power to minimize interference), as the devices that operate in this band are frequently worn close to the body or implanted. There is an ISM band (Industrial, Scientific and Medical, 433.050 to 434.790 MHz) in this frequency range as well. This band allows higher output powers and license free operation, but it is only defined for region 1 (Europe, Africa and Middle East).

1.1.1 Objectives

The proposition of this work consists in the study and the conception of improved RF reflection coefficient measurement systems. To this end, the use of under-sampling down-conversion in the inevitable analog to digital converter and software-defined reflection coefficient measurement will be investigated.

Specific objectives will be:

- Design and implementation of a software defined RF reflection coefficient measurement system based on under-sampling down-conversion for 400 MHz UHF in discrete technology, with software-defined part using standard Fast-Fourier-Transform (FFT) together with complex division.
- Characterization and evaluation of the system with simulations and measurements.

- Due to limitations in the availability of integrated circuit (IC) design software in the second half of the doctorate studies, the third objective was changed from IC design to elaboration of design considerations of the software-defined part of the system including analysis of different signal processing algorithms and a possible matching algorithm.

1.2 Text organization

This text is organized in nine chapters. The second chapter will focus on the theory behind the used techniques for the reflectometer and the according literature review. Chapter three focuses on the principle of operation of the system which was designed to proof the feasibility of the new approach, and chapter four presents computer simulation results of the whole system and some of its parts. The fifth chapter then reports details on the design and implementation process of the system, while measurement results after short-open-load (SOL) compensation are presented in chapter six. Chapter seven describes design considerations for the software defined part of the system, and chapter eight the outline of an improved complex control algorithm for an automatic impedance matching system. And finally, chapter nine presents the conclusion.

2 Literature Research and Theoretic Foundation

2.1 Reflection Coefficient Measurement Techniques

The reflection coefficient Γ is the complex division of the reflected signal V_r by the incident signal V_i: $\Gamma = V_r/V_i$. Thus, the straight forward first step to measure the reflection coefficient is generally to separate the incident and the reflected signal. This can be done by a circulator or by a directional coupler. After separating the two signals, the phase and amplitude relationship between both must be analyzed to allow the complex division. The most direct approach to do this is the Polar one.

2.1.1 Polar Approach

The phase between the reflected and the incident signal can be measured using a phase detector such as a digital XOR gate or phase-frequency-detector (PFD), a Gilbert cell or a passive mixer, that outputs an analog signal proportional to the phase between the two signals (after filtering). For this operation, the amplitude information must be removed from both waves, to make sure the phase detector only measures the phase. This can be done with the help of limiters. This system was described in [17]. For the complete reflection coefficient information, the amplitudes of incident and reflected wave also must be measured, what can be done with diode power detectors. This system was described in [18]. In total, three analog signals must be measured: The phase between reflected and incident signal, the reflected amplitude, and the incident amplitude.

The disadvantage of the diode power detectors is, due to their nonlinear nature, they require some sort of linearization technique in the following circuitry or software, what makes the system more complex. A block diagram of this topology can be found in Figure 2.1. Here, V_r is the reflected signal, V_i is the incident signal, and φ is the phase between both signals.

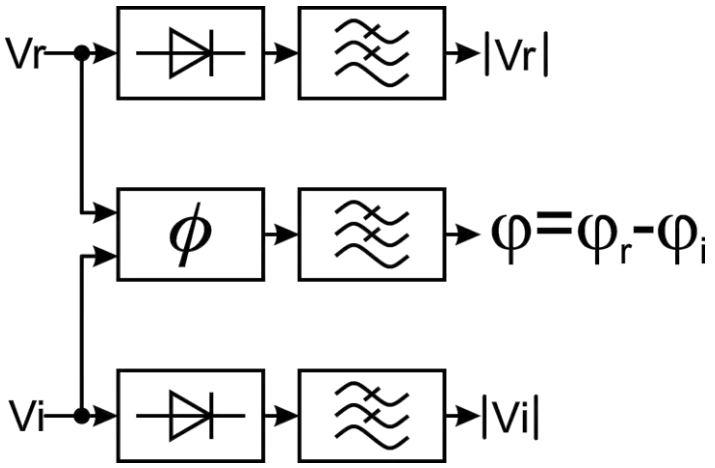

Figure 2.1: Polar Approach

2.1.2 Cartesian Approach

Instead of using diode power detectors, it is possible to use in total three mixers (active or passive), two for the reflected wave and one for the incident wave. This way the phase detector also becomes unnecessary. Additional circuitry is necessary to generate 0° and 90° phase shifted signals with removed amplitude information from the incident signal. For the reflected wave, one mixer is fed with the 0° signal and one mixer is fed with the 90° signal, resulting in linear quadrature amplitude (IQ) demodulation [19]. This means an Vr_I in-phase (the real component of the wave) and a Vr_Q quadrature DC-signal (the imaginary component of the wave) are generated. So, the vector of the wave is known in a Cartesian coordinate system, effectively preserving the amplitude and the phase in respect to the incident signal. Finally, the amplitude of the incident signal must be measured, possibly also with in-phase amplitude demodulation (resulting in Vi_I), this way avoiding diode power detectors and their nonlinearities.

This results in three analog signals that must be measured, like in the Polar case, if the 0° signal is synchronous with the V_i signal

However, in some cases the 0° signal is not completely synchronous to V_i, for example if a quadrature injection locking oscillator is used to generate the 0° and 90° signals [16]. This results in a slowly varying phase shift between V_i and the 0° signal. Nevertheless, the 0° and the 90° signal have the correct, fixed phase relationship required for the IQ-demodulation. In this case a fourth mixer can be added, using the 90° signal also for the incident wave and allowing the removal of the variable phase shift from the final reflection coefficient result.

As these techniques convert the RF signal directly to DC, they suffer from problems with small amplitudes and large offsets of the mixers. A block diagram of this topology can be found in Figure 2.2.

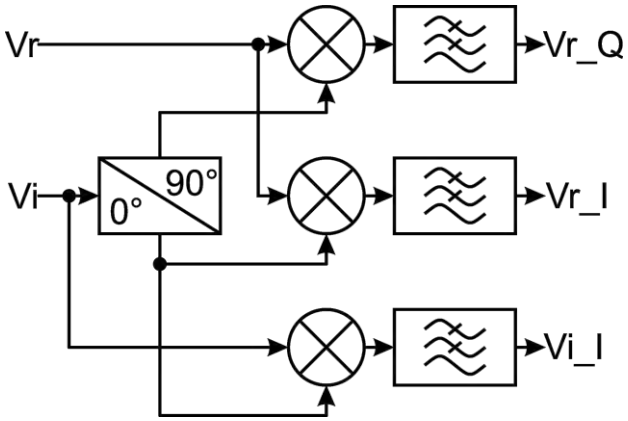

Figure 2.2: Cartesian Approach

2.1.3 Superhet Cartesian Approach

In reference [19], a second approach is described, removing problems with small amplitudes and large offsets of the mixers as seen in the normal Cartesian approach. Basically, the conversion of the signal is not done to DC but to an intermediate frequency (IF), allowing amplification and filtering of the IF more easily while preserving the phase and amplitude information. Afterwards, the IF is quadrature demodulated in a second step,

allowing for much higher amplitudes and so much lower influence of the offset error. This approach is known as Superheterodyne or Superhet. Unluckily it adds a lot of complexity to the system, and still converts the signals to DC in the analog domain in the end, making them sensitive to analog errors and inaccuracies.

2.1.4 New Software Defined Approach proposed in this Thesis

Instead, the two RF signals can be analyzed directly, moving all complexity of the system into the digital domain. The down-conversion to IF can be done directly by under-sampling down-conversion (explained in the subsection of the same name) in the Analog-to-Digital Converter (ADC), removing effectively all mixers from the circuit. Then the IF is analyzed by software, for example incorporating a Fast Fourier Transform (FFT), thus removing all offset and possibly modulation errors completely. This technique should present the best possible performance for such a measurement system.

However, such a software defined system may have limitations in its bandwidth in the case of strong under-sampling or presents high requirements for the digital signal processing due to the high signal frequencies in the case of weak or no under-sampling. Here, reducing the signal processing requirements seemed more important than large bandwidth, therefore relatively strong under-sampling was adopted. In this case, high bandwidth modulated signals may produce erroneous outputs, not only due to problems with synchronizing the sample clock to the input signal.

The sampling clock of the ADC must have a defined relation to the RF frequency. This can be achieved with a (partly analog) Phase-Locked Loop (PLL) frequency synthesizer or with a Direct Digital Synthesis (DDS) circuit. As DDS for RF are still expensive and produce spurs, we will assume that a PLL is used (described in the subsection “PLL for frequency synthesis”).

For all approaches mentioned above, a device to separate the signal incident to the antenna from the signal reflected from the antenna is necessary. This function can be fulfilled by a (quasi-) circulator or by a (bi-) directional coupler. Couplers have advantages of low main-line loss (at the PA output) and better isolation between main-line and the measurement branch of the system.

2.2 Directional Couplers

Directional couplers direct power from the main input port (IN) to the main output port (TRA) and vice-versa, also see Figure 2.4 in section 2.2.3. The attenuation in the main path can be low, often considerably below 4dB [16]. The important property of the directional coupler is that it directs a portion of the incident signal to a third port called “coupled” (CPL) and a portion of the reflected signal to a fourth port, often called “isolated” (ISO), making them available for measurement. The magnitude of the signals at the CPL and ISO ports is often considerably smaller than the one of the signals at the IN and TRA ports. Commercially available are coupling values between 3dB and 50dB [16]. A common coupling value of 15dB means that the signal at the CPL port is 15dB lower than the incident signal at the IN port.

It is possible to integrate directional couplers on a monolithic chip. For frequencies in the range around 30 GHz, these couplers are based on parallel coupled transmission lines on the chip [20]. For frequencies in the UHF-range (300 MHz to 3 GHz), these transmission lines would get long (the typical length is a quarter of the wavelength), so that they are frequently replaced by lumped inductors, coupled inductors or transformers and capacitors for integration. Often the used technology is one specialized in producing high quality passives, such as Integrated Passive Device (IPD) technology, Silicon On Insulator (SOI) or Low Temperature Cofired Ceramic (LTCC).

2.2.1 Lumped Inductors and Capacitors

There exist various topologies, with capacitors as central element [21], or with inductors as central element [22]. There instead of the Pi-topology to replace the transmission lines with lumped components, the T-topology can also be used [23]

2.2.2 Coupled Inductors (Transformers) and Capacitors

Interleaved Transformers have been used [24] as well as coupled synthesized co-planar waveguides (Figure 2.3) [25]. However, on low impedance or lossy substrates coupled inductors are difficult to achieve.

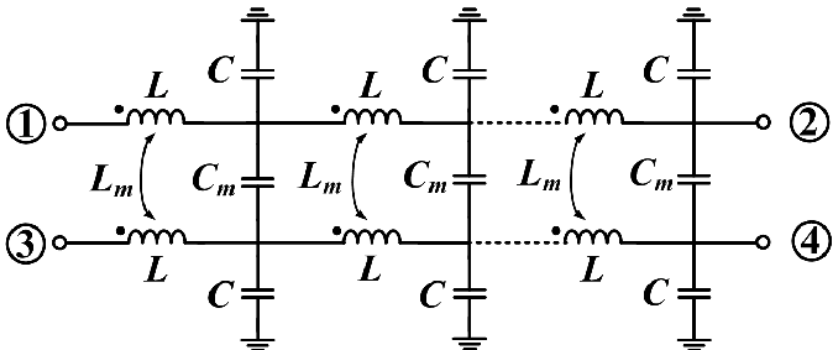

Figure 2.3: Directional Coupler using Coupled Inductors and Capacitors [25]

2.2.3 Two Transformer Approach

The two-transformer approach is widely used for low frequency directional couplers in wire-wound discrete technology. A theoretical treatment of the matter can be found in [26]. However, at least for high impedance / low loss substrates, an integration of this topology for higher frequencies should also be possible. A simplified schematic of this approach is shown in Figure 2.4.

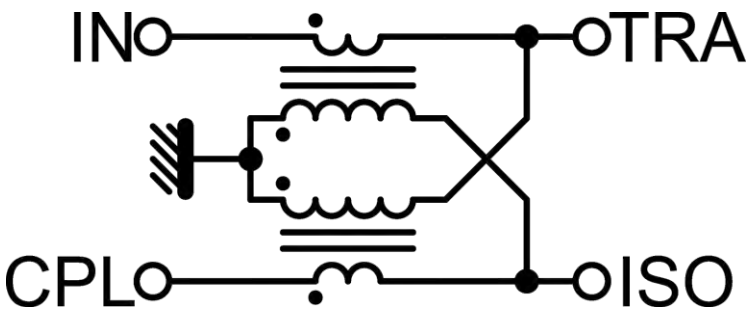

Figure 2.4: Directional Coupler using two Transformers

2.3 Injection Locking Frequency Dividers and their Digital Counterparts

The elaborated system described in chapter 3 uses the main-line frequency as a source of the sampling clock for the ADC, so that the signal frequency and the sample clock have a fixed relationship, what is necessary for the function of the under-sampling down-conversion process. The main-line frequency is converted to the sampling frequency using a PLL circuit that will be described later. However, the used PLL circuit has a limited input frequency range. Therefore, it is necessary to divide the RF before feeding it into the frequency synthesis integrated circuit. Frequency dividers will be discussed in this subsection.

2.3.1 Injection Locking Frequency Dividers

Injection locking (IL) is a nonlinear effect in any oscillator (mechanical, optical, and electronic). An oscillator locks (synchronizes) to an injected (fed in) signal from an external source, if the signal has sufficient amplitude and is inside a frequency range (lock range) around the natural frequency of the oscillator.

By adequate construction of the oscillator and input point of the injected signal, it is also possible to lock the oscillator to low integer fractions n (n=2, 3, …) of its natural frequency. This operation results in a divide-by-n frequency divider.

An example schematic of a divide-by-3 IL frequency divider is shown in Figure 2.5 [27].

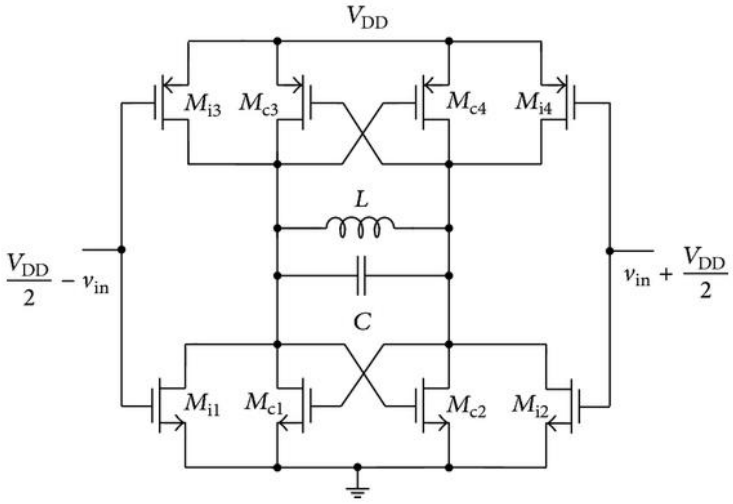

Figure 2.5: Divide-by-3 Injection Locking Frequency Divider

2.3.2 D Type Flip-Flop as Frequency Divider

A D-type flip-flop is a simple digital one-bit memory [28]. Typical implementations store the state of the data (D) input on the rising edge of the clock (CLK) input. There can be two

outputs, the positive Q-output, and the negated /Q-output. However, often only one of these is used. For using the flip-flop as frequency divider, it is practical to connect /Q with D. So on every positive CLK-edge the state of the flip-flop is inverted. This results in a signal of half frequency of CLK at the /Q-output. In other words, it is a divide-by-two frequency divider.

2.3.3 Digital Counters

For higher division ratios, typically digital counters are used [29]. They comprise of flip-flops and combinatory logic. The counter adds one to the digital number stored in its flip-flops on every rising edge of the clock input. When the value stored in a digital register is reached, the counter is reset and the output toggles. This way, the division factor is determined by an (within the limits of the register) arbitrary integer number. The size of the register (and counter) in bits sets the absolute lower limit of the output frequency of the divider.

In the system of section 3, the output signal of the frequency divider is fed into the frequency synthesis circuit, that is described in the following subsection.

2.4 Phase-Locked Loop for Frequency Synthesis

The simplest implementation of a phase-locked loop comprises of phase detector (PD), a loop filter and voltage-controlled oscillator (VCO). The output frequency of the VCO (f_{VCO}) is fed into one of the two inputs of the PD. The other input (REF) is connected to a reference frequency. The PD is often a digital circuit such as a phase-frequency-detector (PFD) with a charge pump (CP) that outputs positive or negative current pulses (i_d) of a duration proportional to the phase difference between its input signals. These current pulses must be converted to a DC-voltage (v_c) that can be fed into the VCO. This function is fulfilled by the loop filter that also serves to stabilize the feedback loop. The output frequency of the VCO thus is controlled by the phase difference between itself and a reference signal. This results in a VCO output of the same frequency and with zero phase difference to the REF input. Adding a frequency divider at the REF input (frequently named R-divider), the output frequency can be divided down in integer steps [30], also see Figure 2.6 in the next section.

2.4.1 Integer-N

The system becomes more interesting by adding a frequency divider into the feedback from VCO to PD. Then the frequency can not only be divided, but also multiplied by an integer factor. This divider is normally named N-divider [30]. The output frequency thus is described by equation (1):

$$f_{VCO} = \frac{N}{R} \cdot f_{REF} \tag{1}$$

The crucial advantage of this approach is that it produces no spurious frequency components at the output (spurs). However, for a fine adjustability of the output frequency, a high division ratio R is necessary, reducing the frequency at the PD significantly. This produces higher phase noise and slower adjustment of the loop to a change in input phase

or frequency because the loop filter needs to have a lower cut-off frequency and thus slower step response. A block diagram of such a system can be found in Figure 2.6. It includes an A divider for the output signal f_{out} for additional frequency flexibility.

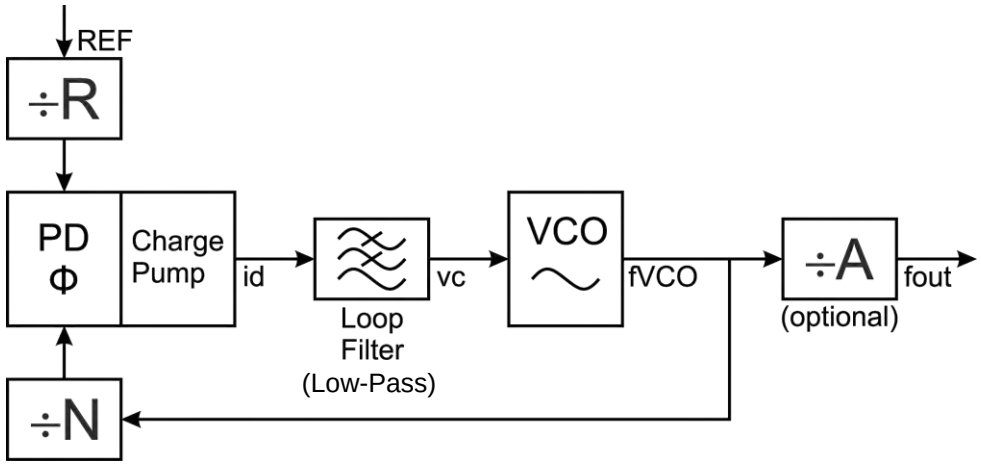

Figure 2.6: Integer-N PLL Frequency Synthesizer

2.4.2 Fractional-N

By switching the N divider between several different values continuously, a fractional multiplication factor can be achieved. While achieving a fine adjustment step size, this makes much higher frequencies at the PD possible, thus making the loop faster and reducing phase noise. The disadvantage of this approach is that it produces spurs at the output due to the continuous switching operation. This problem can be reduced by applying the ΣΔ modulation technique to the switching operation [30].

2.4.3 Loop Analysis

The PFD has a transfer function as in equation (2), with the reference phase θ_{REF} and the output phase θ_{OUT} of the PLL, the conversion factor K_{PFD} and the output current i_d of the PFD:

$$i_d = K_{PFD}(\theta_{REF} - \theta_{OUT}) \tag{2}$$

With (3), including the charge pump current I_{CP}:

$$K_{PFD} = \frac{I_{CP}}{2\pi} \tag{3}$$

The loop filter can be active or passive. For high frequencies, generally a passive filter is used. The most common topology, consisting of the capacitors C_1 and C_2 and the resistor R, is shown in Figure 2.7.

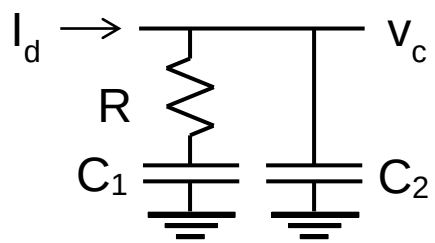

Figure 2.7: Common loop filter topology

The capacitor C_2 is typically chosen so that its impedance is much higher than the impedance of R and C_1 together at the PD-frequency. A typical value is $C_2=C_1/10$. This means that C_2 can be neglected for the analysis of the loop behavior.

The simplified transfer function of the loop filter is shown in equation (4):

$$F(s) = \frac{sC_1R+1}{sC_1} \tag{4}$$

The transfer function of the VCO with the N-divider is (5), with the output phase θ_{VCO}, the control voltage v_c and the conversion factor K_{VCO} of the VCO as well as a loop division factor N:

$$\frac{\theta_{VCO}}{v_c} = \frac{1}{N}\frac{K_{VCO}}{s} \tag{5}$$

The general PLL closed loop gain is (6):

$$\frac{\theta_{VCO}}{\theta_{REF}} = \frac{KF(s)}{s+KF(s)} \tag{6}$$

With (7):

$$\mathrm{K} = \frac{K_{PFD}K_{VCO}}{N} \tag{7}$$

In its general form, the overall feedback loop transfer function becomes (8) for this loop filter, using (4) and a general second order transfer function:

$$\frac{\theta_{VCO}}{\theta_{REF}} = \frac{\omega_n^2\left(\frac{2\zeta}{\omega_n}s+1\right)}{s^2+2\zeta\omega_n s+\omega_n^2} \tag{8}$$

With the natural frequency ω_n (9) and damping factor ζ (10):

$$\omega_n = \sqrt{\frac{K_{PFD}K_{VCO}}{NC_1}} \tag{9}$$

$$\zeta = \frac{R}{2}\sqrt{\frac{K_{PFD}K_{VCO}C_1}{N}} \tag{10}$$

The filter components can be calculated as in (11) to (14):

$$C_1 = \frac{K_{PFD}K_{VCO}}{N\omega_n^2} \tag{11}$$

$$\mathrm{R}= \zeta\frac{2N\omega_n}{K_{PFD}K_{VCO}} \tag{12}$$

$$|X_{C2}| = 10|X_{C1} + R| \tag{13}$$

With the reactances $X_{C1}$and X_{C2} at the natural frequency as follows in (14):

$$X_{Ca} = \frac{1}{j\omega_n C_a} \text{ with a = 1, 2.} \tag{14}$$

The 3dB bandwidth ω_{3dB} of the PLL is then (15):

$$\omega_{3dB} = \omega_n \cdot \sqrt{1 + 2\zeta^2 + \sqrt{4\zeta^4 + 4\zeta^2 + 2}} \tag{15}$$

This equation can be approximated with (16) and (17):

For $\zeta > 1.5$ $$\omega_{3dB} = \omega_n \cdot 2\zeta \tag{16}$$

For $\zeta < 1.5$ $$\omega_{3dB} = \omega_n \cdot \left(1 + \zeta \cdot \sqrt{2}\right) \tag{17}$$

Typical values for ζ are between 0.3 and 5, often used is 0.707.

The condition for loop stability is as in (18) [30], with phase detector frequency ω_{PD}:

$$\frac{\omega_{PD}}{\omega_n} \geq 2\pi\zeta \tag{18}$$

For ζ=0.707, this means that this ratio must be bigger than 4.4. Often used is a ratio of 10.

These considerations should be sufficient to calculate the components of the loop filter for the application. All other necessary equations can be found in the datasheet of the integer-N PLL frequency synthesizer Analog Devices ADF4360-9 that was chosen to be used for the system (see chapter 3.4).

The central principle of the planned system is based on using the output frequency of the frequency synthesis circuit as sample frequency for the analog-to-digital converter (ADC). The ADC performs under-sampling on the two RF signals, this way down-converting them to a low intermediate frequency (IF) in the digital domain.

2.5 Analog-to-Digital Converter (ADC)

Generally, an ADC consists of two main blocks: the sample and hold (or track and hold) block and the quantizer. The information here is a summary of [31], chapter 17.3, therefore no single references are given.

2.5.1 The Quantizer

The quantizer converts the continuously valued (but often discrete time) input signal into discrete digital values. This leads to quantization noise because the signal now is made from steps rather than smooth transitions. The quantization noise content of the digital signal is lower when more discrete steps are used, however the quantization into a larger number of discrete values (represented by a higher number of bits) comes at the cost of lower conversion speed or higher circuit complexity or both. So, a reasonable trade-off must be found.

For one bit (above or below a certain level), the quantization function is fulfilled by an electronic comparator. For higher bit numbers (or more discrete digital values), there are various topologies, each with certain advantages and disadvantages:

Flash-ADC

Mainly consists of a high number of comparators according to the number of levels to be distinguished. Necessary are 2^b-1 comparators for a given number of resulting bits "b". For 8 bits, this means 255 comparators are used, what results in a comparably large integrated circuit. On the other hand, this topology is extremely fast (several giga-samples per second / GSps) The outputs of the comparators are merged to the desired digital number by means of a logic network.

Pipelined ADC

Here smaller flash-ADCs and digital-to-analog converters (DACs) are used in sequence to reduce the number of required comparators considerably. The first stage does a rough estimate, and the second stage refines this estimate. Also, three or more stages are possible. This way, a 10 bit ADC composed of two 5 bit stages just needs 2□(2^5-1) = 2^6-2 = 62 comparators instead of 1023. Disadvantages are a lower (but still high in the order of many MSps) conversion speed and possible challenges with the linearity of the circuit. An according topology is used by the ADC utilized in this work (see section 3.5, especially Figure 3.5).

Successive Approximation Register ADC

It is also possible to use just one comparator together with a DAC and a memory device that stores the last state of the conversion. Then the comparison level of the comparator is varied according to a bisection method: First the whole conversion range is divided in two and the MSB is generated. Then according to the MSB, either the upper or the lower half is again divided in two and checked, generating the second bit, and so forth until the desired number of bits is reached. This requires exactly the same number of conversion cycles as the number of desired bits, thus it is still relatively fast (up to some MSps) and allows higher numbers of bits with much lower circuit complexity than the preceding architectures.

Delta-Sigma ADC (ΔΣ)

For highest resolutions (number of bits up to 24 and beyond), it is common practice to use noise shaping. With the according circuitry, it is possible to shift noise into higher frequency bands of the digital signal where it can be removed using a digital filter. This way, the noise content of the signal of interest is reduced considerably. Unfortunately, the digital filter also reduces the data rate. Even though high sample rates (some MSps) are used, the effective data rate can be only some few Sps at the highest resolution.

An advantage is that the resolution and the data rate can be traded off against each other by adjusting the digital filter, to achieve an optimum compromise for each application. The quantizer here is often just one-bit and combined with an analog summation and integration circuit and a one-bit DAC. The final resolution is achieved only by the digital filtering. However, this scheme relies on massive over-sampling instead of under-sampling and is thus less interesting for this work.

ADC based on Time Measurement

For even lower conversion speeds, there are simpler topologies for ADCs achieving similar resolutions as the $\Delta\Sigma$ and avoiding the digital filter. These are mostly based transferring the level of the signal into an integration time (producing a slope, thus the name "Single-Slope" and "Dual-Slope") and measuring this level dependent time by means of a clocked counter, or in newer setups a Time-to-Digital Converter (TDC). The latter allows a better trade-off between conversion time and resolution, but still the conversion times tend to be longer than on a $\Delta\Sigma$. The main advantage is the lower complexity.

2.5.2 The sample and hold (S&H) or track and hold (T&H) block

This is an analog device which either opens the input only for a short period of time and keeps its output constant at the last input value during the remaining time of a conversion cycle (S&H), or the output follows the input for half the cycle and stays constant at the last value for the remaining cycle (T&H). Usually, an electronic sample switch and a capacitor to hold the signal at its last value, as well as buffer amplifiers are used for these means.

Either way, the conversion to a digital value takes place during the constant output phase. This has the effect of converting the continuous time input into a discrete time (clocked) output signal as input for the quantizer, leading to sampling effects as utilized in this work and described in section 2.6.

2.6 Under-Sampling Down-Conversion

To understand the concept of under-sampling down-conversion, we will first consider general aliasing effects that occur in sampled systems.

2.6.1 Sampling Effects: Aliasing

A periodic sampling scheme always produces aliases. These are artifacts that are spaced symmetrically around integer multiples of the sampling frequency [32].

Generally, a signal can only be reconstructed from samples if the sampling frequency is more than two times the bandwidth of the signal, because otherwise the aliases overlap in the frequency domain. For a low pass signal (baseband, centered at the frequency zero), this means that the sampling frequency must be at least two times the highest frequency of the signal. However, for band pass filtered signals, a much lower sample frequency can be sufficient to reconstruct the signal completely, as shown in Figure 2.8.

The alias frequencies of an input frequency f_i sampled at the sample frequency f_s appear at f_{alias}, with N being any integer number (in theory, the aliases also appear at negative frequencies), as described by equation (19) [32]:

$$f_{alias}(N) = |f_i - f_s N| \tag{19}$$

The lowest alias frequency is for the following N (20):

$$N = \mathrm{FLOOR}\left(\frac{f_i}{fs}\right) \tag{20}$$

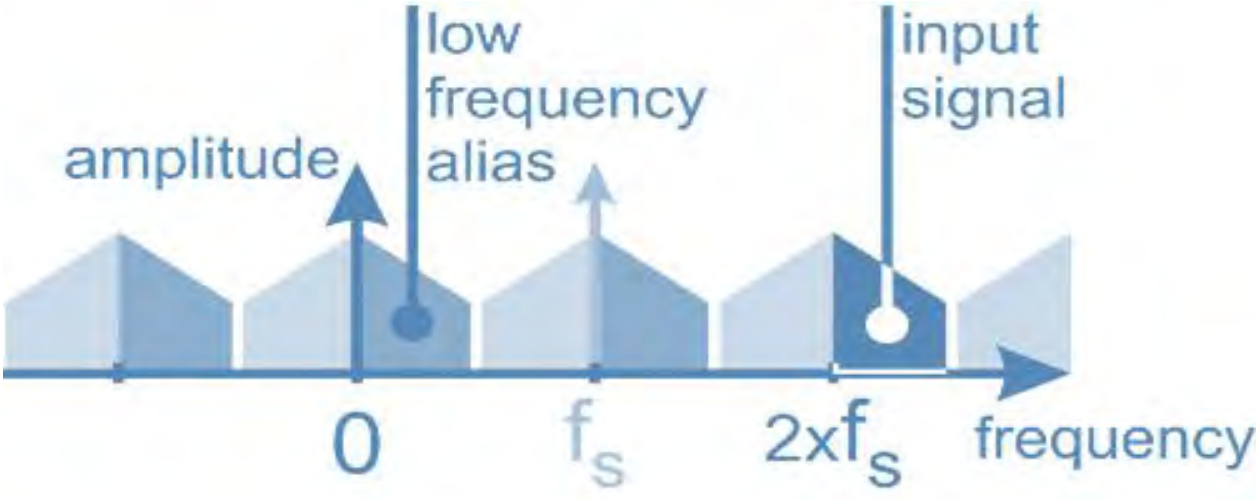

Figure 2.8: Aliasing for Under-Sampling

2.6.2 Under-Sampling

Under-sampling means that 2fi > fs.

The following conditions (21) and (22) [34] must be fulfilled by the sample frequency to prevent overlap of the aliases:

$$\frac{2f_H}{n} \leq f_s \leq \frac{2f_L}{n-1} \tag{21}$$

$$1 \leq n \leq \text{FLOOR}\left(\frac{f_H}{f_H - f_L}\right) \tag{22}$$

With f_H and f_L being the upper and the lower limit of the frequency band, respectively.

The signal to noise ratio in dB of any fast ADC is as in (23) [34] (using the number of bits b of the ADC):

$$SNR_{ADC} = 6.02b + 1.76 + 10\log_{10}\left(\frac{f_s}{2f_H}\right) \tag{23}$$

If f_s is lower than $2 \cdot f_H$, the SNR is reduced. This can also be expressed as a reduction in effective number of bits (ENOB) of the ADC, as shown in equation (24) [34].

$$ENOB = \frac{SNR - 1.76}{6.02} \tag{24}$$

For $f_s/(2f_H) = 1/100$, the effective number of bits is reduced by 3.33 compared to Nyquist sampling.

As can be seen from (23) and (24), strong under-sampling reduces the SNR and the ENOB considerably, and raises the low-noise and high-speed requirements on the ADC and the PLL. A trade-off between the acceptable noise level or the required number of bits and the digital processing speed requirements is necessary.

The phase relationship between different signals sampled synchronously is preserved in this operation, as is shown in Figure 2.9. The thin lines are the input signals, the marked

points are the samples, and the dashed thicker lines are the low-frequency aliases of the input signals. All graphs of the same color belong together. In the inverted aliases (consider Figure 2.8), the phase is inverted, what can easily be corrected in software.

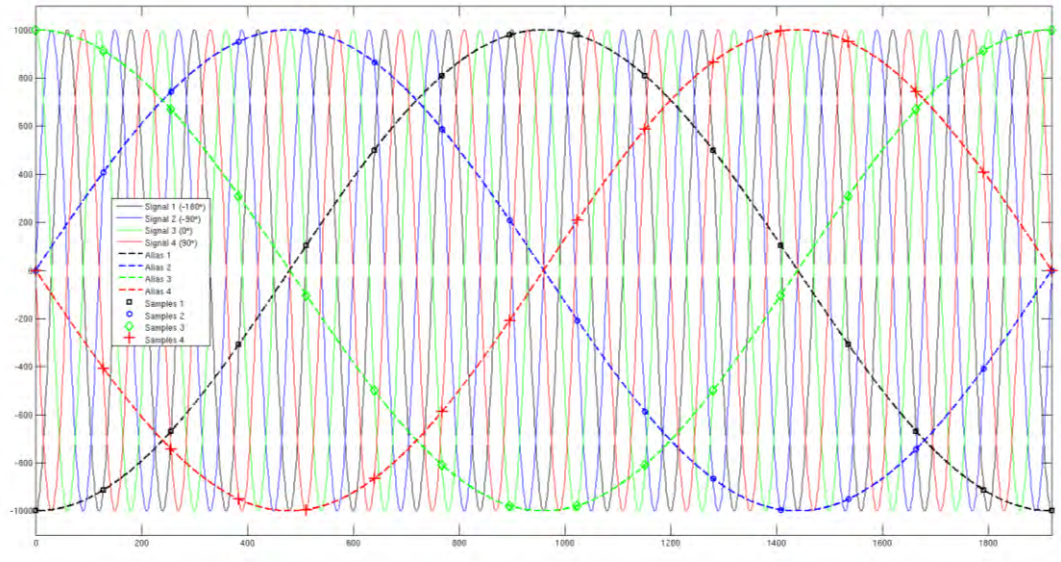

Figure 2.9: Under-Sampled Signals with Different Phase

Example 1:

Applying (19) and (20) to the carrier frequency f_i of 400 MHz and f_s of 7.99 MHz and 15.96 MHz, N becomes 50 and 25, and f_{alias} becomes 0.5 MHz and 1.0 MHz, respectively.

2.6.3 Under-Sampling Down-Conversion

As can be seen from the above example 1, the carrier frequency of 400 MHz is aliased to a much lower frequency in the digital domain. This is equivalent to a mixing operation known as down-conversion. The advantage is that for this operation, only the sampling operation of the ADC is already enough, additional mixers are not required. The disadvantage is that the sampling circuitry of the ADC must have enough bandwidth for the highest signal frequency f_H, not only for f_{alias}. This means the bandwidth of the S&H device of the ADC must be more than 400 MHz for the example. This makes the ADC more complex and thus expensive.

2.7 Fast Fourier Transform (FFT)

The Fourier Series describes any continuous time periodic signal by a series of sine and cosine functions and their harmonics as well as exponentials [32]. Aperiodic continuous time signals are described accordingly by the Fourier Transform, mathematically transforming the time-domain signals into the frequency domain. However, the signal in the digital domain is now discrete time, discrete valued. Such signals can be transformed into the frequency domain by applying the Discrete Fourier Transform (DFT). There exist various algorithms exploiting properties of subsets of digital signals (such as defined signal lengths and the like) to gain great advantages in computation speed, all together known as Fast Fourier Transform (FFT).

2.7.1 Frequency Bins

As stated above, the FFT breaks down a discrete time signal into a defined set of sine and cosine functions (discrete frequencies). Effectively, as the frequencies of these functions are already defined relative to the record length of the discrete time signal, the transform only calculates the amplitudes for these functions. For each calculated signal frequency (called frequency bin) there exists an amplitude for a sine and for a cosine function, together forming the complex amplitude at this frequency. Phase shifts in the signal contents are depicted in Cartesian form in the complex amplitude, however the Polar form can be derived using the arc tangent function (ATAN2 in most computational systems) for the phase and the square root of the sum of squares of the two complex amplitude components for the magnitude according to the usual way of converting Cartesian into Polar form of complex variables.

2.7.2 Radix-2, Radix-4, Radix-8

The most well-known FFT algorithm (Cooley–Tukey algorithm) is Radix-2, this means that the input data must have a record length of 2^n points for the algorithm to work, n being a positive integer number. The algorithm then uses a divide-and-conquer approach to reduce the number of necessary computations. Accordingly, additional speed can be gained for signals with 4^n (Radix-4) or 8^n (Radix-8) points, allowing for breaking down the signal even more effectively [35]. Other algorithms such as split radix and many more are known and can be found in the respective literature.

3 Principle of Operation

3.1 System Block Diagram

The measurement system comprises of the following blocks, as shown in the block diagram in Figure 3.1:

a) Directional Coupler
b) Pre-Divider
c) Integer-N PLL Frequency Synthesizer
d) Fast Under-Sampling-Capable Analog-to-Digital Converter (ADC)
e) System Processor
f) Microcontroller Software
g) Personal Computer Software

Not shown are impedance matching networks.

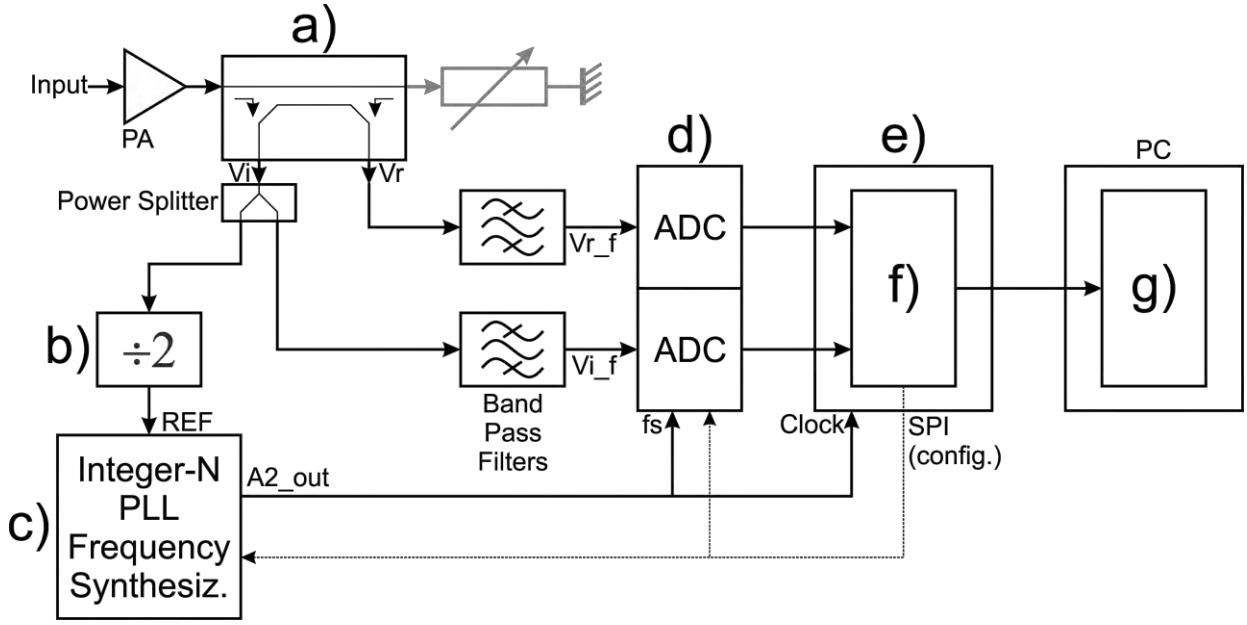

Figure 3.1: System Block Diagram

The incident signal V_i as obtained by the directional coupler a) is distributed to two different blocks via a resistive power splitter.

One of the two following blocks is the frequency pre-divider b) which divides the V_i-frequency by two for adjusting it to the input range of the integer-N PLL frequency synthesizer c). The output frequency of the PLL is then used as sample clock for the fast under-sampling-capable analog-to-digital converter (ADC) d) and as data transmission clock for the parallel synchronous data link to the system processor e).

The second portion of V_i is fed into one of two anti-aliasing band-pass filters and from there into one ADC input. Also, the reflected signal V_r from the directional coupler is fed into the second anti-aliasing band-pass filter and into the second ADC input. The converted data from the ADC is transferred to the system processor via a parallel synchronous data link.

The system processor microcontroller software f) then analyzes the digital signals and calculates the reflection coefficient. Finally, the results can be fetched by the personal computer (PC) software g) via an USB link.

The PC software also configures the PLL via the system processor and a serial peripheral interface to the PLL chip.

The Power Amplifier (PA) and the directional coupler a) are necessary for the operation of the system but not included in the system board. This allows the operation with different PA and coupler devices and thus makes testing different configurations simple. Similarly, the system controller was not placed on the system board. Instead, a ready-made microcontroller development board was used and connected to it by means of an adapter board and a ribbon cable. This saved development time and makes the system more flexible.

3.2 Directional Coupler

For a discrete setup in the 400 MHz UHF range, various bi-directional couplers can be used. An example is the Mini-Circuits SMA bi-directional coupler ZFBDC20-62HP-S [36]. The advantages of this device are its bi-directionality and the comparably low main-line loss (0.25dB at 400 MHz), as well as a coupling factor of about 20dB at 400 MHz. However, as stated before, different couplers can be attached to the system board for testing, what can be interesting for following projects where micro-couplers can be developed.

3.3 Pre-Divider Block

The Pre-Divider has the purpose of adjusting the relatively high input frequency of the incident wave signal to the capabilities of the input of the Analog Devices ADF4360-9 integer-N PLL frequency synthesizer (please refer to section 3.4).

An all-digital 74AUP1G80 D-type flip-flop [37] was used due to its high operation frequency capability, its Schmitt-Trigger inputs, and its CMOS nature and case size both of which simplified the application into the system.

This device has the maximum frequency f_{max} ratings depending on the load capacitance C_L as shown in Table 3.1. This load capacitance is assumed to be connected from the output of 74AUP1G80 to ground.

Table 3.1: Max. frequency vs. load capacitance of 74AUP1G80 at 3.0 to 3.6 V supply

C_L (pF)	f_{max} (MHz)
5	619
10	550
15	481
30	309

The D input of the 74AUP1G80 has an input capacitance of 1.5 pF, the REF_in capacitance of the ADF4360-9 frequency synthesizer is 5 pF maximum. Therefore, with a total load capacitance around 6.5 pF, the Pre-Divider should be able to operate faster than 550 MHz at 3.0 V supply and around 25°C. The clock input of the 74AUP1G80, like the D input, is a

Schmitt-trigger input (it has two trigger voltages V_H and V_L and hysteresis). So, the input signal can be DC biased to $(V_H+V_L)/2$ without damage to the device. For Vcc=3.0 V this is about V_{bias}=1.5 V.

For neglectable effect on the impedance matching circuit, the coupling capacitor C2 can have a minimum value of 150 pF (100 times the input capacitance of the 74AUP1G80) and a maximum value of 1.5 nF (time constant limited), and the bias resistors R4 and R5 between 10 kΩ and 100 kΩ (same considerations). An additional protection resistor R7 of 68 Ω in series with the clock input is needed because the input voltages after the inevitable impedance matching can go considerably beyond the supply rails. According to the Keysight Advanced Design System (ADS)-impedance matching tool, the impedance matching can consist of a 91 nH series inductor L1 close to the source and a 270 fF parallel capacitor C1 close to the input of the device. In fact, this parallel capacitor can be manufactured using the stray capacitance of the PCB. The phase jitter contribution of the bias resistors is about 0.3 ps RMS for the 100 kΩ case (according to simulations). A circuit diagram of the impedance matching circuit is shown in Figure 3.2. The node "IMP_MAT_IN" is connected to the power splitter at the V_i output of the directional coupler.

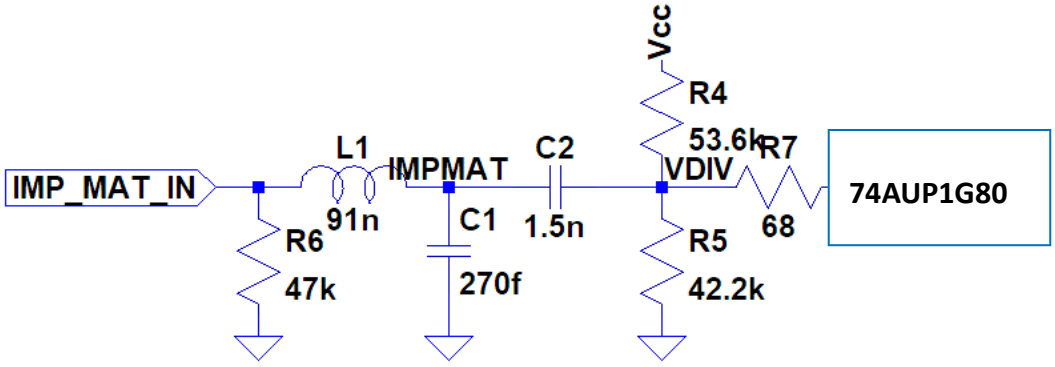

Figure 3.2: Impedance Matching & Bias Circuit before Pre-Divider

3.4 Integer-N PLL Frequency Synthesizer

The purpose of the PLL Frequency Synthesizer is to generate the sample frequency for the ADC from the output signal of the pre-divider. It should be an integer-N device to avoid spurious frequencies in the output that might interfere with the under-sampling down-conversion. Furthermore, it should allow the generation of frequencies in the desired sample frequency range from UHF frequencies and allow software configuration of all important parameters. This narrows down the field of available PLL circuits considerably, as today most PLL ICs are either fractional-N or do not support the required input and output frequencies especially the additional A divider at the output is interesting in this case. The chosen Analog Devices ADF4360-9 [38] is a compromise, as it allows only 250 MHz maximum reference frequency (REF) what makes a pre-divider at this input necessary. The integrated voltage-controlled oscillator (VCO) supports frequencies between 65 and 400 MHz, and the device is configured via the SPI interface, which is common in modern microcontrollers such as the system processor.

As shown in Figure 3.3, there are three main frequency dividers that can be programmed separately. The R counter divides the REF input frequency by values between 1 and 16383.

After that, it is fed into the phase frequency detector (PFD), that supports frequencies up to 8 MHz. The output of the VCO is divided by the B counter by values between 3 and 8191. The result is also fed into the PFD. The frequency of the VCO is also divided by the A counter that can assume values between 2 and 31. As the output of the A counter generally has a non-50% duty cycle, there exists a further divide-by-2 divider A_2 that can be activated if necessary to ensure 50% duty cycle. The output frequency f_{out} for the system is taken from the A_2 divider. There are more options to configure this device, but these are the most important ones.

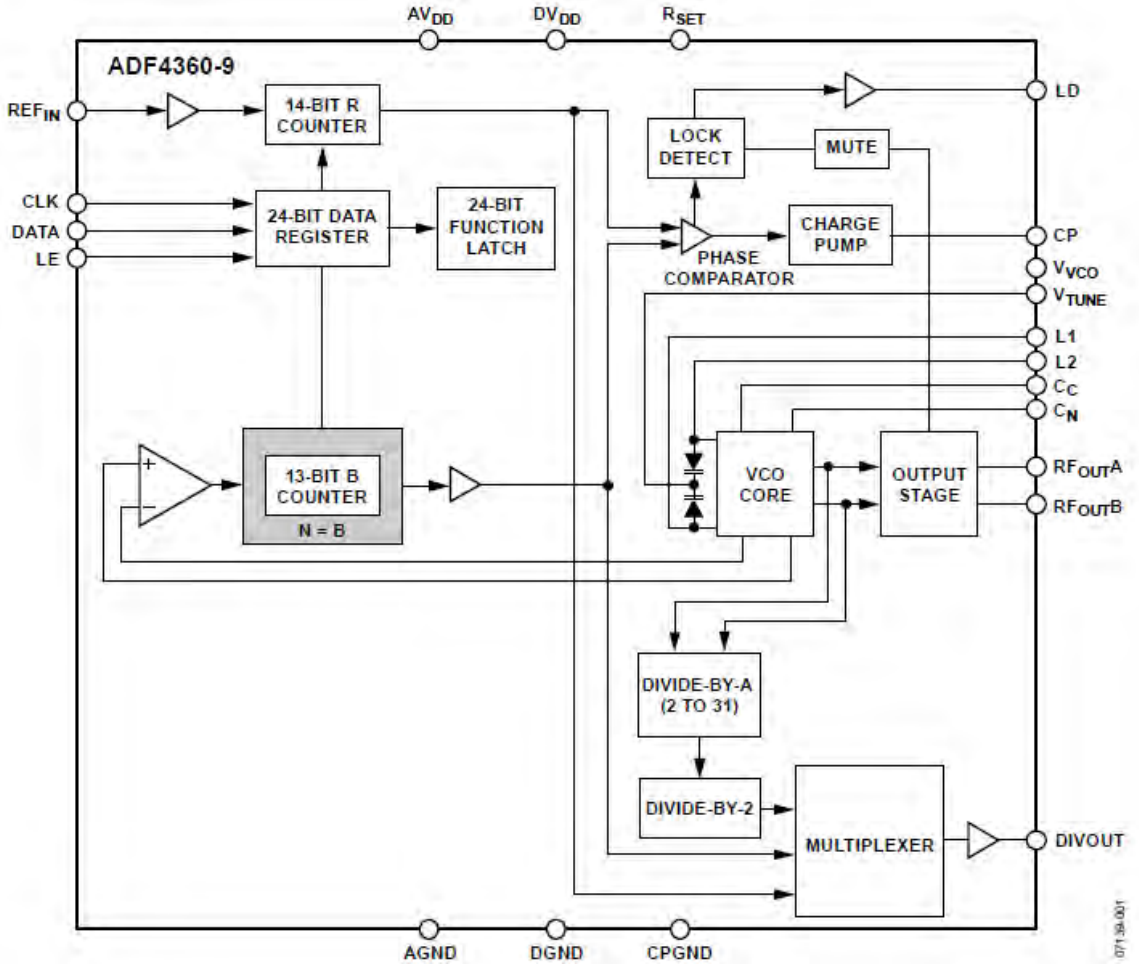

Figure 3.3: ADF4360-9 Functional Block Diagram [38]

Table 3.2 shows some useful first design values for the application in the under-sampling down-conversion system.

Table 3.2: Chosen values for ADF4360-9 frequency synthesizer and effect

Value	Setting 1	Setting 2	Unit	Equation | Info
f_{PreDiv}	400.0	400.0	MHz	Signal frequency at pre-divider input
f_{REF}	200.0	200.0	MHz	= f_{PreDiv} / 2 | Frequency at PLL input
R	625	625	-/-	R-divider value of PLL
f_{PFD}	320.0	320.0	kHz	= f_{REF} / R | Frequency at PFD
B	798	799	-/-	B-divider of PLL (normally called N)
f_{VCO}	255.36	255.68	MHz	= Bf_{PFD} | VCO output frequency
A	**8**	**16**	-/-	A-divider value of PLL
A_2	2	2	-/-	A2-divider for 50% duty cycle
f_{out}	**15.96**	**7.99**	MHz	= f_{VCO} / (AA_2) = f_s | Sample frequency
f_{alias}	1.0	0.5	MHz	= $f_{ADC} - f_{out}$FLOOR(f_{ADC} / f_{out}) | Alias freq.

It would also be possible to derive the REF signal from a quartz, for example of 10 MHz. Then no pre-divider is necessary. But in case the input frequency of the analog-to-digital converter f_{ADC} changes, this translates directly to a higher relative error. For 5 MHz difference, the relative error is greater than 90%. So, it is advisable to derive f_{REF} via a pre-divider from f_{ADC}. This way the relative error stays constantly at the value shown in Table 3.2. This is because the relationship between the alias frequency f_{alias} (the signal generated in the digital domain by the under-sampling down-conversion) and the frequency that is assumed to be frequency-slot 2 of the FFT is fixed. This relationship will be analyzed in 3.4.1.

3.4.1 Calculation of Good PLL Counter Values

To begin with, the equation (19) as stated earlier (repeated here for convenience) describes the relationship between the input frequency f_i, the sample frequency f_s and the resulting alias frequency f_a, where N is a positive integer that selects the corresponding alias:

$$f_a(N) = |f_i - f_s N| \tag{19}$$

This includes all aliases, the folded (inverted) as well as the non-inverted ones.

The inverted aliases are selected using (25):

$$f_a(N) = f_s N - f_i \tag{25}$$

As we are only interested in the non-inverted aliases, we can apply (26):

$$f_a(N) = f_i - f_s N \tag{26}$$

As we are converting the sampled signal using a fast fourier transform (FFT) algorithm, ideally the alias frequency falls directly within one of the bins of the FFT. This means that there must be at minimum an integer relationship W (samples per aliased wave) between f_s and f_a like shown in (27):

$$W = \frac{f_s}{f_a} \tag{27}$$

Ideally, W is equal to the FFT length, or 1/2 or 1/3 of it. Higher division ratios may result in reduced accuracy. As the FFT length is a power of 2 (in our case 32, but could be 64), W of 16, 32 and 64 are especially interesting.

We can easily express f_a depending on f_s and W (28):

$$f_a = \frac{f_s}{W} \tag{28}$$

Combining (26) with (28) to eliminate f_a this results in (29), that allows to calculate possible sample frequencies:

$$f_s = \frac{f_i}{\frac{1}{W}+N} \tag{29}$$

As the sample frequency is generated by an integer-N phase-locked loop (PLL), it is interesting to know if it is possible to generate these sample frequencies exactly using an integer-N. As it turns out, it is, if the reference frequency f_r of the PLL is also derived from f_i (30):

$$f_r = \frac{f_i}{2} \tag{30}$$

The integer-n PLL equation was presented in (1), what is repeated here for convenience (just f_{VCO} was replaced by f_s already), with C_N being the n counter value and C_R being the reference counter value, both integer:

$$f_s = f_{VCO} = f_r \frac{C_N}{C_R} \tag{1}$$

Using (1), (29) and (30), the following expression (31) is found:

$$\frac{C_N}{C_R} = \frac{2W}{NW+1} \tag{31}$$

From this, we see the following integer relationships (32) and (33):

$$C_N = 2W \tag{32}$$

$$C_R = NW + 1 \tag{33}$$

Furthermore, in the case of the ADF4360-9 PLL, the following equation (34) is valid for the values of the so-called A and B counters C_A and C_B, given that the real division ratio of the PLL output is $C_A 2$ (when the additional A_2 divider for 50% duty cycle is active):

$$C_N = \frac{C_B}{2C_A} \tag{34}$$

With this relationship, the actual register values for allowable voltage-controlled oscillator (VCO) frequencies can be calculated.

Please note the non-integer sample frequencies resulting from this technique. Some values for f_s and C_R depending on N and W for an input frequency of 400 MHz are shown in Table 3.3:

Table 3.3: Some Values for fs and C_R depending on N and W for fi=400 MHz

N	W					
	16 (C_N=32)		**32** (C_N=64)		**64** (C_N=128)	
	f_S (MHz)	C_R	f_S (MHz)	C_R	f_S (MHz)	C_R
4	98,461538	65	99,224806	129	99,610895	257
8	49,612403	129	49,805447	257	49,902534	513
16	24,902724	257	24,951267	513	24,975610	1025
32	12,475634	513	12,487805	1025	12,493899	2049
40	9,984399	641	9,992194	1281	9,996095	2561
50	7,990012	801	7,995003	1601	7,997501	3201

3.4.2 Loop Filter

Finally, the external inductors L_{EXT} for the VCO can be calculated as in (35) [38]:

$$f_0 = \frac{1}{2\pi\sqrt{9.3\ pF(0.9\ nH+L_{EXT})}} \tag{35}$$

This results in $L_{EXT} \approx 41$ nH for the necessary center frequency f_0 of the VCO of 255 MHz.

The K_{VCO} can be obtained for this L_{EXT} from the diagram in Figure 3.4 as about 6 MHz/V. Equation (35) and the diagram can be found on page 20 of the datasheet [38].

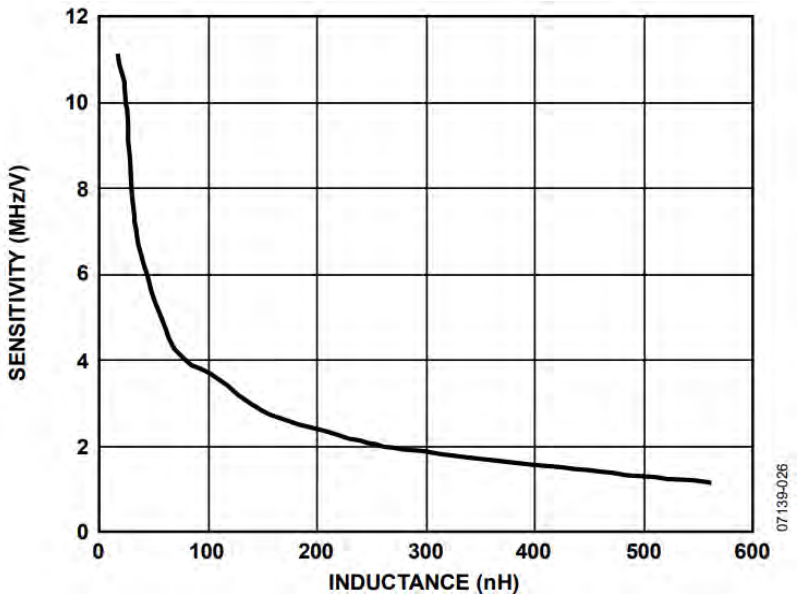

Figure 3.4: Diagram of VCO sensitivity from [38]

With the help of the Analog Devices ADIsimPLL PLL simulation software, the loop filter components were defined to C_1 = 817 pF, C_2 = 169 pF and R = 14.7 kΩ (the names of C_1 and C_2 are switched in the software, and R is called R_1), and L_{EXT} = 40.3 nH. The simulated phase jitter was about 3.5 ps RMS and the time until lock detect 50 µs (power up transient, so first lock to 7.99 MHz after configuration). However, the simulations show that it is advisable to

wait another 50 µs (so 100 µs in total) before using the PLL to measure, because before this time the phase still oscillates to values higher than the phase noise.

3.5 Fast Analog-to-Digital Converter (ADC)

The ADC converts the analog signal to the digital domain by a sampling procedure. It is necessary to sample two signals (the incident and the reflected signal from the directional coupler) simultaneously, to be able to calculate the reflection coefficient correctly. The RF signal is under-sampled with a fixed ratio, generating an alias frequency in the digital domain. This procedure preserves the phase relationship between the two input signals.

The input sample and hold circuitry of Linear Technology LTC2296C [39] ADC has a full power bandwidth of 575 MHz, so under-sampling of a 400 MHz RF signal is possible. This ADC converts the signal with a maximum sample rate of 25 Msps and a resolution of 14 bits, divided in six pipelined stages as shown in Figure 3.5.

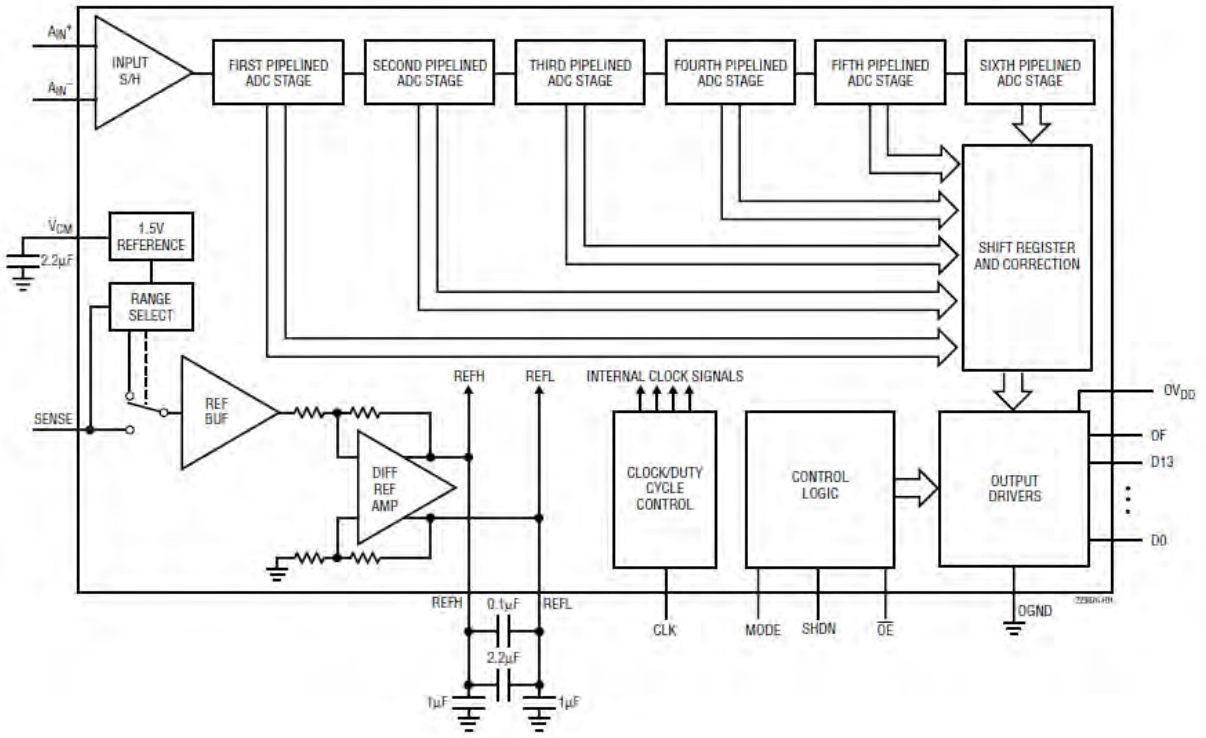

Figure 3.5: Functional Block Diagram LTC2296C (1 channel of 2) [39]

According to the SNR at Nyquist sampling given at the data sheet [39], the LTC2296C has an ENOB of 12.03 bit. An under-sampling factor of around 50 was necessary due to the chosen system processor and its parallel transmission capabilities as discussed below. A reduction of 3.33 bits in the ENOB, as it occurs with an under-sampling factor of 50, thus yields an ENOB of 8.7 bits. In other words, the error due to the reduced SNR is below 0.25% of the full-scale input of the ADC and can be further reduced by averaging. This is sufficiently low when compared with the other error sources in the signal patch such as the anti-aliasing bandpass filters, the baluns and the impedance matching at the ADC input and the like.

Therefore, this ADC meets the requirements of the application for a comparably low price. Furthermore, Linear Technology was known for their high-quality ADCs, and pin-compatible ADCs for higher sample rates are available, in case techniques to relieve the system processor of the full (or double) sample rate transmission speed are applied in a later design, such as introducing a first-in-first-out (FIFO) memory into the system. However, for a first design, it is good practice to avoid adding too much functionality and overcomplicating the system.

As stated before, in a system without a separate FIFO, the sample rate that can be achieved (and thus the under-sampling factor) depends on the system processor, as the sample rate and the data clock rate are equivalent in full parallel data transmission mode (31 digital signal pins are needed). In the also supported multiplexed mode the effective data clock is double the sample rate, but only 16 digital signal pins are needed. If the processor is not able to record the data fast enough, the system will not work properly. For reasons described in the next section, a sample rate of 7.99 MHz for the multiplexed mode and 15.96 MHz for the full parallel mode was chosen (see also Table 3.2, f_{out}). These frequencies produce useful aliasing frequencies with the system frequency of 400 MHz for further processing in the system processor.

3.6 System Processor

The system processor has two main purposes: It records the values from the fast ADC, and it executes the measurement software modules (and optionally also the control software modules). The modern STM32F4 family was chosen for the reflectometer because it offers a hardware single precision floating point unit as well as some hardware digital signal processing (DSP) capabilities for a low price, both of which makes it interesting for this work. From this family, the STM32F446RE (F446) was chosen due to its high possible clock frequency. It is an STMicroelectronics ARM Cortex-M4 microcontroller with digital signal processing (DSP) extensions that supports 180 MHz clock frequency (225 DMIPS) [40], [41]. As a microcontroller of this family clocked with 80 MHz (100 DMIPS) reaches 10 MHz data clock for the parallel communication [42], it is expected that a parallel transmission speed of more than 20 MHz should be possible (using the internal RAM). To save pins, the multiplexed option of the ADC was used, making the effective data clock double the sample rate. To stay under 20 MHz in this condition, the sampling clock of the ADC should be below 10 MHz.

The NUCLEO-F446RE printed circuit board (PCB) [43] including this microcontroller was used. This PCB has sufficient pins free for the application and makes them accessible via 0.1” headers. It also already includes all the necessary infrastructure to run and debug the microcontroller for a low price.

For transferring the parallel data fast enough, the direct memory access (DMA) controller 1 (DMA1) inside the F446 [44] was used. It is triggered by input pins of general purpose timer 2 or 5 (TIM2 or TIM5, for example TIM2CH1 and/or TIM2CH2 on PA0 and PA1, respectively) and copy the state of all PB and/or PC pins to memory, automatically

incrementing the address in memory for each copy action, until enough data-sets for the further processing (16 or 32) are stored in memory, and then stop and disable the copying automatically until it is re-enabled by software.

The SPI1 is used to communicate with the attached PLL device ADF4360-9.

The UART2 (PA2 & PA3) is used for communication with the host computer for data recording via the ST-LINK/V2-1 USB debug adapter virtual com port. The ST-Link is part of the Nucleo-PCB but not of the microcontroller.

A block diagram of the described modules can be found in Figure 3.6. V_r is the reflected signal and V_i the incident signal as output by the directional coupler.

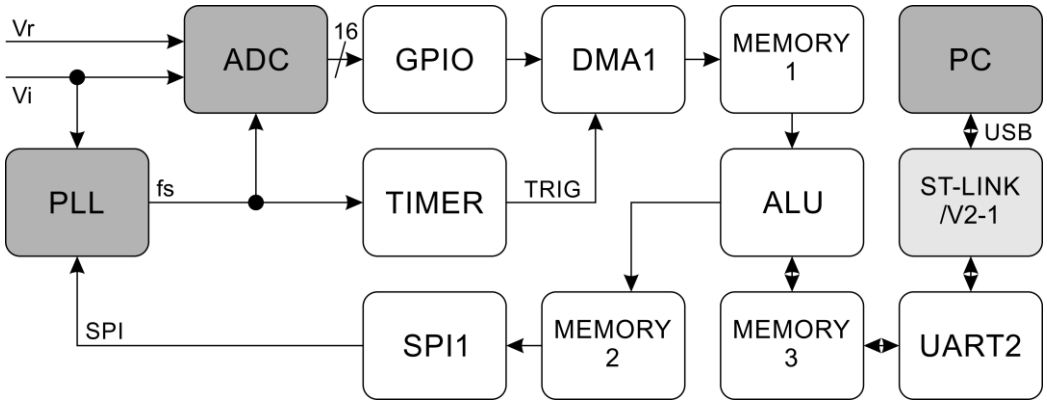

Figure 3.6: Involved microcontroller modules (white) and connected hardware (grey)

3.7 Microcontroller Software

The purpose of the microcontroller software is to provide connectivity with ADC and PLL, to record and process the ADC signals and to send the resulting data to the PC. It configures the timers, the DMA1 and the input port pins for the operation described in the chapter before.

It also configures the SPI1 and the UART2 for communication with other devices and the personal computer. Additionally, it implements device drivers for the ADF4360-9 and the LTC2296C to configure or read out these devices.

3.7.1 Real Fast-Fourier Transform

Finally, the microcontroller software implements a method for calculating complex wave information from the ADC sampled values and to calculate the reflection coefficient based on this information. This involves the use of an ARM Cortex-M4 optimized real Fast Fourier Transform (rFFT) algorithm from the Cortex Microcontroller Software Interface Standard (CMSIS) DSP library [45]. This algorithm (arm_rfft_fast_f32) calculates complex frequency data from real input data based on a mixed-radix algorithm ("multiple radix-8 stages are performed along with a single radix-2 or radix-4 stage…"). For this purpose, "the real sequence is initially treated as if it were complex to perform a CFFT. Later, a processing stage reshapes the data to obtain half of the frequency spectrum in complex format".

While rFFT frequency-bin 1 contains two real values (offset and the Nyquist frequency bin packed together), especially the frequency-bins 2 (for 16 samples record length) or 3 (for 32 samples record length) are of interest. They contain the complex signal amplitudes at the desired alias frequency f_{alias}.

3.7.2 Reflection Coefficient Calculation

This complex data can be used to derive phase and amplitude information, or directly to calculate the reflection coefficient using equations (36) to (38), where **a** and **c** are the real parts and **b** and **d** are the imaginary parts of the complex amplitude of $\mathbf{V_r}$ (reflected signal) and $\mathbf{V_i}$, (incident signal) respectively:

$$V_r = a + j \cdot b \tag{36}$$

$$V_i = c + j \cdot d \tag{37}$$

$$\Gamma = \frac{a \cdot c + b \cdot d}{c^2 + d^2} + j \cdot \frac{b \cdot c - a \cdot d}{c^2 + d^2} \tag{38}$$

For writing the software, application notes and example software for the fast-parallel communication (with the ADC) [42] and for the PLL device driver were collected, and a free IDE (Atollic TrueSTUDIO [46]) that supports the device (and the NUCLEO-F446RE evaluation board) was used. Also employed was the newest version of STM32CubeMX [47], that helped with the generation of the hardware abstraction layer (HAL) drivers for the internal peripherals of the microcontroller. As the peripherals of modern microcontrollers and their application become more and more sophisticated, it is common practice to use such code generators for their configuration.

3.8 Personal Computer software

The personal computer (PC) host software records, evaluates and displays the reflection coefficient data sent to it via the virtual COM port of the ST-LINK/V2-1 debug adapter. It also opens a way to save the data permanently. It was written in C#.

4 Simulations

This chapter will focus on presenting simulation results of the system obtained using MathWorks MATLAB [48] for gaining information on the sensitivity of FFT to frequency variation, Keysight Advanced Design System (ADS) [49] for a characterization of a simplified system model, and Linear Technology LTSpice [50] for characterization of a more complete and thus realistic model of the system. The results of the simulations using Analog Devices ADISimPLL Software for the PLL design [51] are added in Appendix A.1.

4.1 Test of Sensitivity of FFT to Frequency Variation

As the system is software defined, the functioning of the algorithm to calculate the complex amplitude information from the time domain data is crucial. For a first approach, a Fast-Fourier-Transform (FFT) algorithm is applied for this purpose. As a floating-point implementation avoids precision and rescaling pitfalls that might occur when using a fixed point implementation, and as the used microcontroller features a single precision floating point processor, the best algorithm available for the reflectometer in the Cortex-M Software Interface Standard (CMSIS) Digital Signal Processing (DSP) Library [45] is a single precision float-number based algorithm. The MATLAB simulations of the algorithm using a standard MATLAB-FFT were tailored to use single precision float numbers for comparability with the CMSIS-FFT-algorithm implemented on the microcontroller.

For testing the sensitivity of the standard FFT to small variations in the real signal frequency while not adjusting the FFT frequency bin centers, first a signal was generated using the cosine function. Then, the signal was analyzed by the FFT of MATLAB to obtain the complex amplitudes of each signal frequency bin. This information was then compared with the original settings used to generate the cosine signal. The FFT was limited to a small number of bins (in this case 16) as it is the case in the microcontroller implementation.

The comparison was done for various values of phase of the input signal and some frequencies, in the range of fractions of percent up to five percent different from the FFT frequency bin. Finally, the relative errors of the complex amplitude measurement were calculated in percent. One example for the results is presented in Table 4.1. A frequency deviation of 0.1% is equal to 400 kHz when a 400 MHz input signal is applied. This is close to the maximum modulation depth that can be used with the proposed system, due to the bandwidth of the anti-alias band-pass filters. However, this consideration holds for a system with weak under-sampling, where the alias frequency is not as low as it is in the proposed system. In the proposed system, the effect would be amplified if there was no synchronization between the sample clock and the signal frequency, as the signal is down-converted to about 0.5 or 1 MHz, while the deviation would stay 400 kHz. So, the frequency deviation would be in the range of 80% or 40%, respectively. This shows how crucial the synchronization really is, as even at weak under-sampling there is already a strong effect.

The overview of the obtained results is presented in Table 4.2. Relative errors were calculated using (39) and (40), using the ideal/input (Ci) and the simulated/output (Cs) complex signal amplitude and the magnitude of the ideal signal amplitude (MAGi), respectively. The error magnitude MAGe was calculated by (41).

$$\mathrm{Re}\{ErrRel\} = \frac{\mathrm{Re}\{Cs\} - \mathrm{Re}\{Ci\}}{MAGi} \tag{39}$$

$$\mathrm{Im}\{ErrRel\} = \frac{\mathrm{Im}\{Cs\} - \mathrm{Im}\{Ci\}}{MAGi} \tag{40}$$

$$MAGe = \sqrt{\mathrm{Re}\{ErrRel\}^2 + \mathrm{Im}\{ErrRel\}^2} \tag{41}$$

Table 4.1: Example for output of FFT test (here for 0.1% frequency deviation)

FFT Input		**FFT output**		**Expected output**		**Errors**		
PHAi (°)	MAGi	Re{Cs}	Im{Cs}	Re{Ci}	Im{Ci}	Re{ErrRel}	Im{ErrRel}	MAGe
-180	1000	-1000.6	-3.0	-1000.0	0.0	-0.06%	-0.30%	0.31%
-165	1000	-965.6	-261.7	-965.9	-258.8	0.03%	-0.29%	0.29%
-150	1000	-865.0	-502.5	-866.0	-500.0	0.11%	-0.25%	0.27%
-135	1000	-705.5	-709.1	-707.1	-707.1	0.17%	-0.20%	0.26%
-120	1000	-497.7	-867.2	-500.0	-866.0	0.23%	-0.12%	0.26%
-105	1000	-256.4	-966.2	-258.8	-965.9	0.25%	-0.03%	0.25%
-90	1000	2.9	-999.5	0.0	-1000.0	0.29%	0.05%	0.29%
-75	1000	261.6	-964.7	258.8	-965.9	0.28%	0.12%	0.31%
-60	1000	502.5	-864.1	500.0	-866.0	0.25%	0.19%	0.31%
-45	1000	709.1	-704.6	707.1	-707.1	0.20%	0.25%	0.32%
-30	1000	867.8	-497.2	866.0	-500.0	0.18%	0.28%	0.33%
-15	1000	967.1	-255.6	965.9	-258.8	0.12%	0.33%	0.35%
0	1000	1000.6	3.0	1000.0	0.0	0.06%	0.30%	0.31%
15	1000	965.6	261.7	965.9	258.8	-0.03%	0.29%	0.29%
30	1000	865.0	502.5	866.0	500.0	-0.11%	0.25%	0.27%
45	1000	705.5	709.1	707.1	707.1	-0.17%	0.20%	0.26%
60	1000	497.7	867.2	500.0	866.0	-0.23%	0.12%	0.26%
75	1000	256.4	966.2	258.8	965.9	-0.25%	0.03%	0.25%
90	1000	-2.9	999.5	0.0	1000.0	-0.29%	-0.05%	0.29%
105	1000	-261.6	964.7	-258.8	965.9	-0.28%	-0.12%	0.31%
120	1000	-502.5	864.1	-500.0	866.0	-0.25%	-0.19%	0.31%
135	1000	-709.1	704.6	-707.1	707.1	-0.20%	-0.25%	0.32%
150	1000	-867.8	497.2	-866.0	500.0	-0.18%	-0.28%	0.33%
165	1000	-967.1	255.6	-965.9	258.8	-0.12%	-0.33%	0.35%

Table 4.2: Overview of FFT test results

Input f_dev	Relative Error Magnitude Average	St_dev	Factor MAGe/f_dev
0.00%	0.01%	0.00%	-/-
0.05%	0.14%	0.02%	2.90
0.10%	0.30%	0.03%	2.96
0.25%	0.74%	0.09%	2.95
0.50%	1.49%	0.19%	2.97
1.00%	2.97%	0.37%	2.97
2.50%	7.41%	0.91%	2.97
5.00%	14.80%	1.80%	2.96

It becomes clear that the FFT results have a strong dependency on the input frequency deviation f_dev. In fact, the obtained errors in the measurement of the complex amplitude using the FFT are always about a factor of three times the input frequency deviation. So the considerations regarding the synchronization of sample clock and input frequency presented in section 3.4 are really crucial, even more, as this was a simplified simulation with weak under-sampling.

4.2 Characterization of a Simplified Model of the System using ADS

As first experiments with the PLL implementation in Keysight ADS were not successful due to stability problems with the models that could not be resolved, it was decided to assume the correct PLL output frequency and simulate it using a normal pulse voltage source to trigger the Sampler devices that ADS offers. The inputs of the samplers were set to 50 Ω impedance. After the two Samplers, two Quantizers with 2^{14} steps and a range of ±2V were used to simulate the quantization noise effect of the ADC (that is assumed to have 14 bit and an input range according to the LTC2296C used in the proposed system). The resulting output voltages were analyzed using the function VspecTran. The sample frequency and the VspecTran frequency were adjusted to the respective values based on section 3.4.1. A transient simulation was chosen to be as close to the real-world behavior as possible. The VspecTran simulates the software part (FFT) of the proposed system, here only one bin is used, not the whole spectrum.

On the input side, the same directional coupler model as in [16] was used, this time as single coupler without the connection as bidirectional coupler. This simplified coupler model has the effect that the reflected signal as seen at the sampler inputs has a 180° phase shift. This was corrected by an additional transmission line length at the measurement port output (added numerically to TL1). This transmission line was also used to simulate different phases of reflection coefficients, together with adjusting the magnitude of the reflection coefficients using different values of the terminating resistance R2 at its end. The complete model is shown in Figure 4.1, the simulation results are presented in Table 4.3.

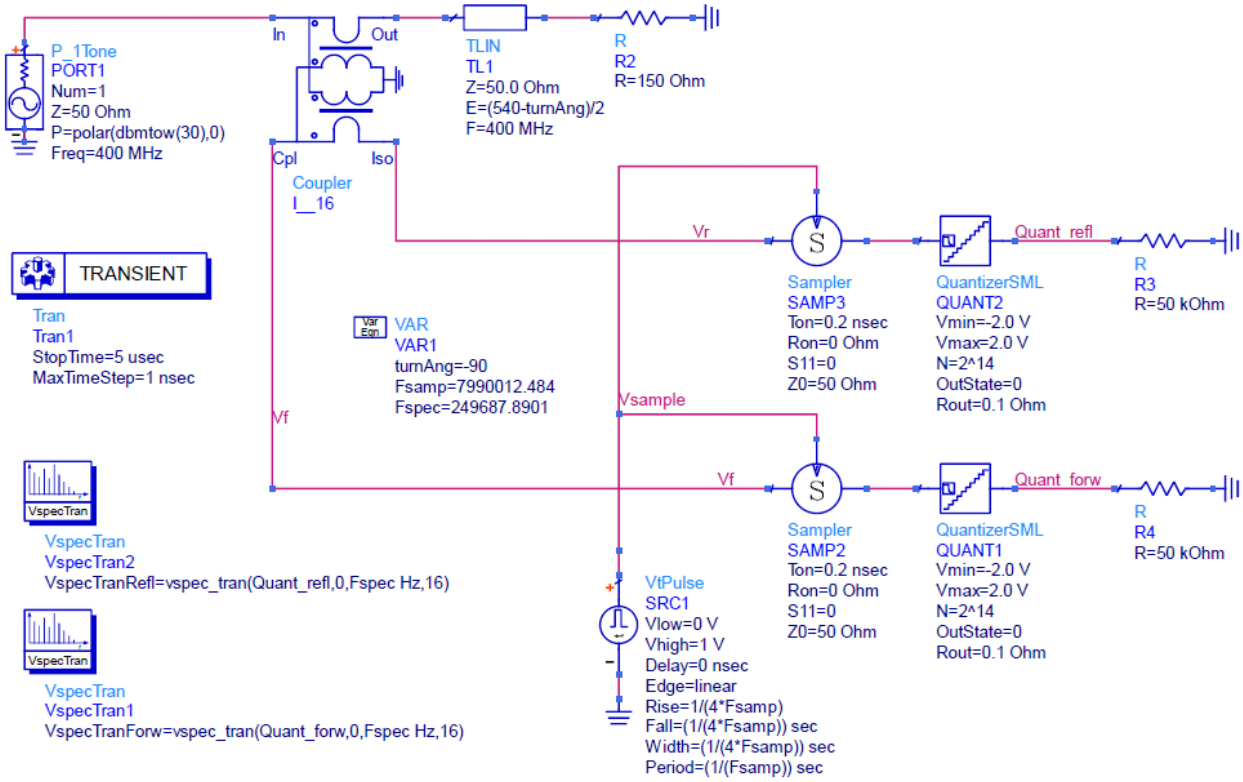

Figure 4.1: Model for the simplified ADS simulations of the whole system

Table 4.3: Results of ADS-Simulations of whole simplified System

Conditions			ReflCoef **Expected**		ReflCoef **Simulated**		**Error** (Simulated-Expected)		
Power (dBm)	Imped. (Ohm)	Phase (°)	Real	Imag	Real	Imag	Real	Imag	Mag
10	50	0	0.0000	0.0000	-0.0148	0.0003	-0.0148	0.0003	1.48%
10	150	-135	-0.3536	-0.3536	-0.3555	-0.3412	-0.0020	0.0124	1.25%
10	150	-90	0.0000	-0.5000	-0.0142	-0.4822	-0.0142	0.0178	2.28%
10	150	-45	0.3536	-0.3536	0.3262	-0.3402	-0.0274	0.0134	3.05%
10	150	0	0.5000	0.0000	0.4649	0.0006	-0.0351	0.0006	3.51%
10	150	45	0.3536	0.3536	0.3237	0.3392	-0.0299	-0.0144	3.32%
10	150	90	0.0000	0.5000	-0.0151	0.4781	-0.0151	-0.0219	2.66%
10	150	135	-0.3536	0.3536	-0.3520	0.3375	0.0016	-0.0161	1.62%
10	150	180	-0.5000	0.0000	-0.4908	0.0003	0.0092	0.0003	0.92%
10	150000	-135	-0.7066	-0.7066	-0.6957	-0.6818	0.0110	0.0248	2.72%
10	150000	-90	0.0000	-0.9993	-0.0138	-0.9636	-0.0138	0.0357	3.83%
10	150000	-45	0.7066	-0.7066	0.6664	-0.6793	-0.0403	0.0273	4.87%
10	150000	0	0.9993	0.0000	0.9442	0.0013	-0.0551	0.0013	5.51%
10	150000	45	0.7066	0.7066	0.6618	0.6781	-0.0449	-0.0285	5.32%
10	150000	90	0.0000	0.9993	-0.0153	0.9557	-0.0153	-0.0437	4.63%
10	150000	135	-0.7066	0.7066	-0.6888	0.6743	0.0178	-0.0323	3.69%
10	150000	180	-0.9993	0.0000	-0.9667	0.0006	0.0327	0.0006	3.27%

Conditions			ReflCoef **Expected**		ReflCoef **Simulated**		**Error** (Simulated-Expected)		
Power (dBm)	Imped. (Ohm)	Phase (°)	Real	Imag	Real	Imag	Real	Imag	Mag
30	50	0	0.0000	0.0000	-0.0149	0.0000	-0.0149	0.0000	1.49%
30	150	-135	-0.3536	-0.3536	-0.3555	-0.3411	-0.0019	0.0124	1.26%
30	150	-90	0.0000	-0.5000	-0.0144	-0.4821	-0.0144	0.0179	2.29%
30	150	-45	0.3536	-0.3536	0.3260	-0.3400	-0.0275	0.0136	3.07%
30	150	0	0.5000	0.0000	0.4650	0.0007	-0.0350	0.0007	3.50%
30	150	45	0.3536	0.3536	0.3235	0.3391	-0.0300	-0.0144	3.33%
30	150	90	0.0000	0.5000	-0.0150	0.4782	-0.0150	-0.0218	2.65%
30	150	135	-0.3536	0.3536	-0.3521	0.3374	0.0014	-0.0161	1.62%
30	150	180	-0.5000	0.0000	-0.4910	0.0002	0.0090	0.0002	0.90%
30	150000	-135	-0.7066	-0.7066	-0.6957	-0.6818	0.0109	0.0248	2.71%
30	150000	-90	0.0000	-0.9993	-0.0139	-0.9637	-0.0139	0.0357	3.83%
30	150000	-45	0.7066	-0.7066	0.6664	-0.6796	-0.0402	0.0271	4.85%
30	150000	0	0.9993	0.0000	0.9442	0.0014	-0.0551	0.0014	5.51%
30	150000	45	0.7066	0.7066	0.6614	0.6778	-0.0452	-0.0288	5.36%
30	150000	90	0.0000	0.9993	-0.0152	0.9557	-0.0152	-0.0436	4.62%
30	150000	135	-0.7066	0.7066	-0.6890	0.6744	0.0177	-0.0322	3.67%
30	150000	180	-0.9993	0.0000	-0.9665	0.0004	0.0329	0.0004	3.29%

The minimum input power was assumed to 10dBm, as this was expected to still work with the pre-divider component 74AUP1G80 in the final system. This component has a hysteresis at its input and thus will not trigger when the amplitude is too low. The maximum power was chosen as 1W (30dBm), because this a common output power for mobile power amplifiers. The impedance values were chosen to yield defined, well distributed reflection coefficient magnitudes (50Ω is matched, therefore magnitude 0, 150Ω yields 0.5, and 150000 is close to 1.0).

4.2.1 Statistics over ADS Simulation Results

Some statistics over input power, reflection coefficient magnitude and phase can also be interesting and are therefore presented in Table 4.4. It can be observed that there is a strong influence of magnitude and phase of the load reflection coefficient on the resulting reflection coefficient measurement error, but no influence of the input power. The latter is desirable for large signal operation of the PA. High reflection coefficient magnitudes generate higher errors, and a phase close to 0° generates less accurate results than a phase close to 180°. Both results are probably due to the influence of the directional coupler, what was investigated more thoroughly in the LTSpice simulations presented in section 4.3.

Table 4.4: Statistics of ADS-Simulations

Pow (dBm)	**Avg Err Γ**	Mag |Γ|	**Avg Err Γ**	Pha Γ (°)	**Avg Err Γ**
10	**3.17%**	0.0	**1.49%**	-135	**1.99%**
30	**3.17%**	0.5	**2.33%**	-90	**3.06%**
		1.0	**4.23%**	-45	**3.96%**
				0	**4.51%**
				45	**4.33%**
				90	**3.64%**
				135	**2.65%**
				180	**2.09%**

4.3 Characterization of a More Complete Model using LTSpice

First experiments with PLL simulation using LTSpice were successful (unlike with ADS), so it was decided to construct a more complete model of the system using this simulation software. The modules of the system were developed separately and connected after being tested.

The module of the Pre-divider (Figure 4.2) already includes the impedance matching and bias circuit, the Schmitt-trigger functionality, and the D-type flipflop used to divide the frequency by two. The PLL model (Figure 4.3) includes the phase detector (with output adjusted to match the behavior of the ADF4360-9 PLL IC), the loop filter (also according to the actual PLL filter), the VCO (with K factor adjusted to the one of the ADF4360-9 when using the chosen inductors) and the dividers (using the chosen values).

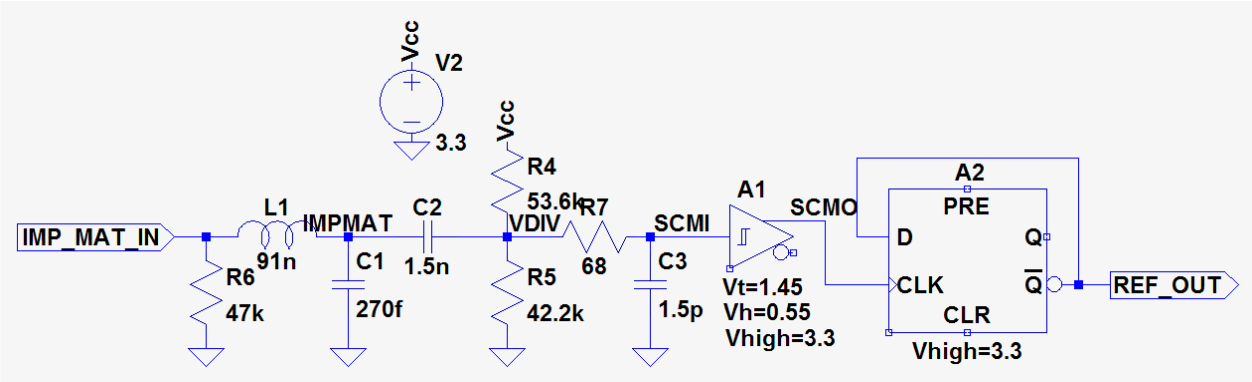

Figure 4.2: Model of Pre-Divider including Input Impedance Matching

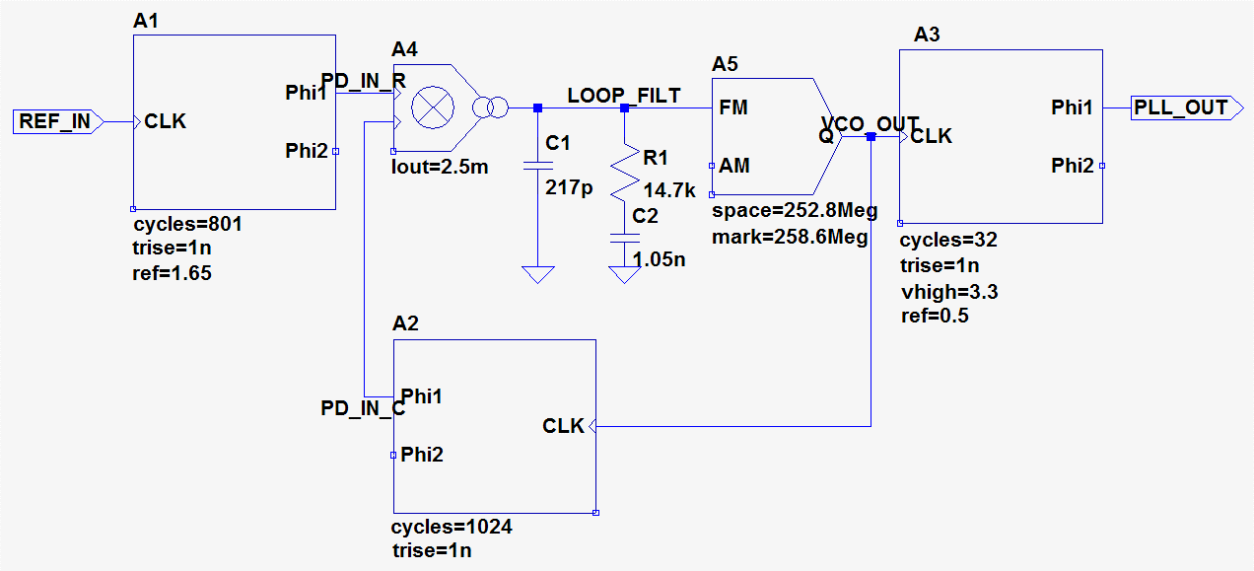

Figure 4.3: Model of the PLL (Adjusted to Calculated Values)

The ADC inputs of the real ADC are differential, thus the refined model (Figure 4.4) includes the necessary baluns and the impedance matching circuitry as well as the input impedances of the real ADC, together with the samplers and two instances of a proprietary behavioral quantizer model using a function-controlled voltage source resembling the 14 bit quantization of the real ADC. The model of the bidirectional coupler (Figure 4.6) was built using two simple couplers (Figure 4.5) in anti-serial connection. The simple couplers have a winding ratio of 10 to show a coupling factor of -20dB and a sufficiently high inductance to work well at 400 MHz, in agreement with the real directional coupler. The model of the Surface Acoustic-Wave (SAW) band-limiting filters was not working correctly (it was instable), so for these simulations it was not included. It would only make a difference if there were more than one frequency at the input. Therefore, this simulation model will not produce reliable results for modulated input.

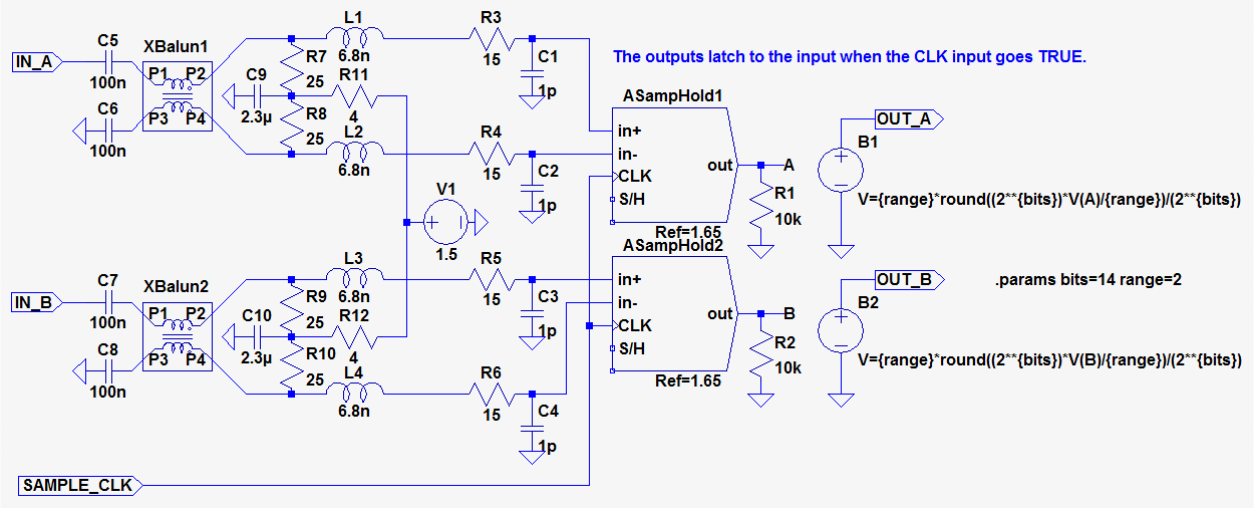

Figure 4.4: Model of the ADC including Input Circuitry for Frequencies > 300 MHz

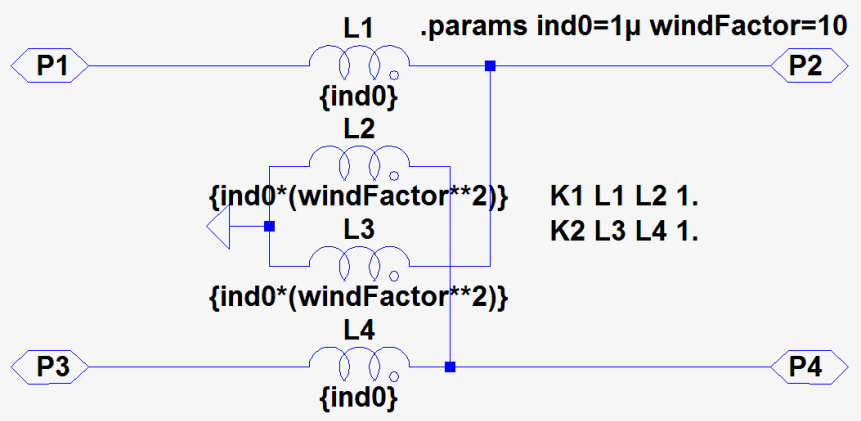

Figure 4.5: Model of single Directional Coupler for Bidirectional Coupler Model

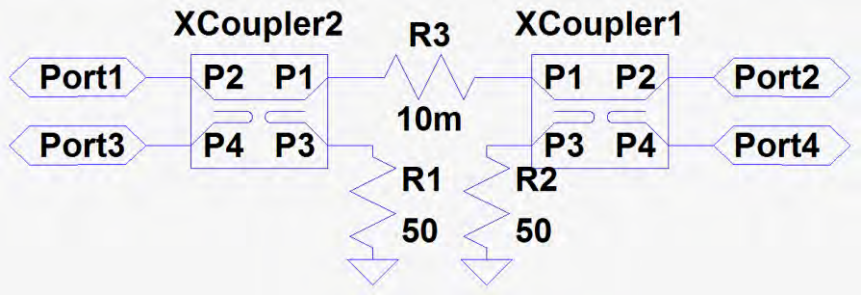

Figure 4.6: The Bidirectional Coupler Model built using two single Couplers

For testing the simulation model, different configurations of the whole system according to the flow-chart in Figure 4.7 were simulated to evaluate the influence of different parts of the system. The pre-divider model and the PLL model were always present as shown above, they were not modified.

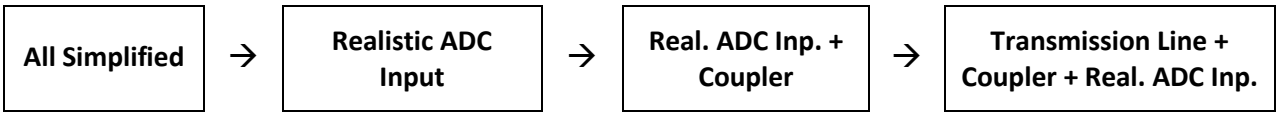

Figure 4.7: Flow-Chart of Simulation Configurations

First, a simplified ADC model using just the sampler and the quantizer was used, and no directional coupler (the expected input voltages were generated using voltage sources, the one for the "reflected" input including a variation of phase and amplitude according to the desired reflection coefficient. As this performed as expected, it was possible to proceed to updating the ADC with the shown model (Figure 4.4) that includes the input circuitry, but still no coupler was used. After testing this setup and obtaining the expected behavior, the coupler was applied but the forward and reflected signals at the coupler main inputs (port 1 and port 2) were generated using voltage sources like before, just at 20dB higher power levels. The signal of the forward voltage source as transmitted through the coupler was absorbed at the impedance of the reflected voltage source and vice versa. The statistics of the results of all these tests are compared in Table 4.6, it would consume too much space here to list all the single values.

Finally, there was only one voltage source (forward) used to generate the input power and a defined reflection coefficient was applied using a defined impedance and transmission line delay, as shown in Figure 4.8. The coupler measurement signal outputs were directed to the according paths via resistive power splitter and attenuator, for incident and reflected signal, respectively. The resistive power splitter was needed as a setup using transmission lines would have become too large for the planned printed circuit board (PCB) setup, and a simple interconnection would have caused undesired reflections in the UHF part of the system. As the resistive splitter introduces attenuation into the incident path, an according attenuator was introduced into the reflected path to keep both paths in balance.

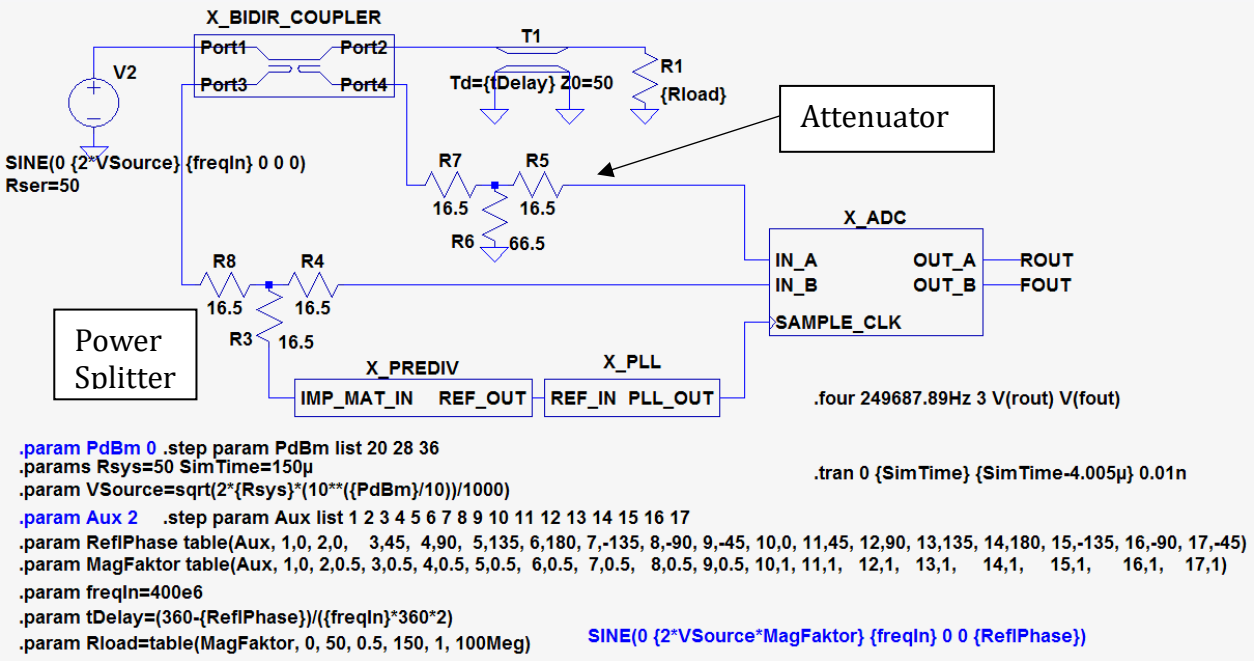

Figure 4.8: Complete Model as used in Final LTSpice Simulation

The simulation results of the final simulation (using the transmission line as shown in Figure 4.8) are presented in Table 4.5. The other simulation results will be shown in the simulation statistics, as here they add best to understanding the influence of different key components. Because of the properties of the implementation of the Fourier analysis ".four" in LTSpice, it was possible to do a complete simulation run with three input power levels and 17 test impedances (so 51 simulations altogether) in one turn. Results of the Fourier analysis are written into text files by LTSpice, which were imported into Excel with an appropriate macro. The reflection coefficient results were calculated in Excel, as this is basically only a complex division and is expected to be executed in the microcontroller with sufficient (single float) precision. For the sake of completeness, a diagram with the expected and obtained reflection coefficients is given in Figure 4.9. In Table 4.5, AUX is an auxiliary number encoding the magnitude and phase condition. The actual condition can be seen in the lines ".param ReflPhase …" and ".param MagFaktor …" in Figure 4.8. For example, AUX 7 encodes a magnitude of 0.5 and a phase of -135°.

Frequency variations were not considered here, because in the proposed system, the frequency of the sample clock is synchronized with the input signal frequency. However, for

fast changes, the PLL might not be able to follow, or if the variation is larger than the bandwidth of the anti-alias band-pass filter, the ADC would also not be able to convert the signals. This however is a separate question and can be addressed in a future project.

Table 4.5: Simulation Results of Complete System using LTSpice

Conditions		ReflCoef **Simulated**		ReflCoef **Expected**		ReflCoef **Error**		
Pin (dBm)	AUX	Real	Imag	Real	Imag	Real	Imag	Mag
20	1	-0.0048	0.0001	0.0000	0.0000	-0.0048	0.0001	0.48%
20	2	0.4876	-0.0064	0.5000	0.0000	-0.0124	-0.0064	1.40%
20	3	0.3474	0.3441	0.3536	0.3536	-0.0062	-0.0095	1.13%
20	4	-0.0009	0.4906	0.0000	0.5000	-0.0009	-0.0094	0.94%
20	5	-0.3487	0.3499	-0.3536	0.3536	0.0049	-0.0037	0.61%
20	6	-0.4948	0.0066	-0.5000	0.0000	0.0052	0.0066	0.85%
20	7	-0.3579	-0.3414	-0.3536	-0.3536	-0.0043	0.0121	1.29%
20	8	-0.0134	-0.4904	0.0000	-0.5000	-0.0134	0.0096	1.65%
20	9	0.3372	-0.3523	0.3536	-0.3536	-0.0163	0.0013	1.64%
20	10	0.9825	-0.0127	1.0000	0.0000	-0.0175	-0.0127	2.16%
20	11	0.6976	0.6892	0.7071	0.7071	-0.0095	-0.0179	2.03%
20	12	0.0015	0.9814	0.0000	1.0000	0.0015	-0.0186	1.87%
20	13	-0.6894	0.7013	-0.7071	0.7071	0.0177	-0.0058	1.86%
20	14	-0.9827	0.0130	-1.0000	0.0000	0.0173	0.0130	2.16%
20	15	-0.7069	-0.6789	-0.7071	-0.7071	0.0002	0.0282	2.82%
20	16	-0.0205	-0.9799	0.0000	-1.0000	-0.0205	0.0201	2.88%
20	17	0.6800	-0.7093	0.7071	-0.7071	-0.0271	-0.0022	2.72%
28	1	-0.0049	0.0002	0.0000	0.0000	-0.0049	0.0002	0.49%
28	2	0.4876	-0.0064	0.5000	0.0000	-0.0124	-0.0064	1.39%
28	3	0.3478	0.3438	0.3536	0.3536	-0.0058	-0.0098	1.14%
28	4	0.0002	0.4913	0.0000	0.5000	0.0002	-0.0087	0.87%
28	5	-0.3473	0.3499	-0.3536	0.3536	0.0063	-0.0036	0.73%
28	6	-0.4951	0.0066	-0.5000	0.0000	0.0049	0.0066	0.82%
28	7	-0.3564	-0.3410	-0.3536	-0.3536	-0.0029	0.0126	1.29%
28	8	-0.0118	-0.4903	0.0000	-0.5000	-0.0118	0.0097	1.53%
28	9	0.3380	-0.3525	0.3536	-0.3536	-0.0156	0.0011	1.56%
28	10	0.9826	-0.0129	1.0000	0.0000	-0.0174	-0.0129	2.16%
28	11	0.6981	0.6898	0.7071	0.7071	-0.0090	-0.0173	1.95%
28	12	0.0048	0.9799	0.0000	1.0000	0.0048	-0.0201	2.07%
28	13	-0.6887	0.6993	-0.7071	0.7071	0.0184	-0.0078	2.00%
28	14	-0.9828	0.0132	-1.0000	0.0000	0.0172	0.0132	2.17%
28	15	-0.7089	-0.6793	-0.7071	-0.7071	-0.0018	0.0278	2.79%
28	16	-0.0229	-0.9807	0.0000	-1.0000	-0.0229	0.0193	3.00%
28	17	0.6818	-0.7078	0.7071	-0.7071	-0.0253	-0.0006	2.53%
36	1	-0.0049	0.0001	0.0000	0.0000	-0.0049	0.0001	0.49%
36	2	0.4876	-0.0064	0.5000	0.0000	-0.0124	-0.0064	1.39%
36	3	0.3458	0.3444	0.3536	0.3536	-0.0078	-0.0092	1.20%
36	4	0.0015	0.4920	0.0000	0.5000	0.0015	-0.0080	0.81%
36	5	-0.3479	0.3513	-0.3536	0.3536	0.0056	-0.0022	0.60%

36	6	-0.4951	0.0067	-0.5000	0.0000	0.0049	0.0067	0.83%
36	7	-0.3578	-0.3412	-0.3536	-0.3536	-0.0042	0.0124	1.31%
36	8	-0.0138	-0.4917	0.0000	-0.5000	-0.0138	0.0083	1.61%
36	9	0.3370	-0.3536	0.3536	-0.3536	-0.0166	0.0000	1.66%
36	10	0.9826	-0.0129	1.0000	0.0000	-0.0174	-0.0129	2.16%
36	11	0.6966	0.6911	0.7071	0.7071	-0.0105	-0.0160	1.92%
36	12	0.0055	0.9852	0.0000	1.0000	0.0055	-0.0148	1.58%
36	13	-0.6889	0.7001	-0.7071	0.7071	0.0182	-0.0070	1.95%
36	14	-0.9826	0.0130	-1.0000	0.0000	0.0174	0.0130	2.17%
36	15	-0.7091	-0.6828	-0.7071	-0.7071	-0.0020	0.0243	2.43%
36	16	-0.0196	-0.9827	0.0000	-1.0000	-0.0196	0.0173	2.61%
36	17	0.6811	-0.7098	0.7071	-0.7071	-0.0260	-0.0026	2.61%

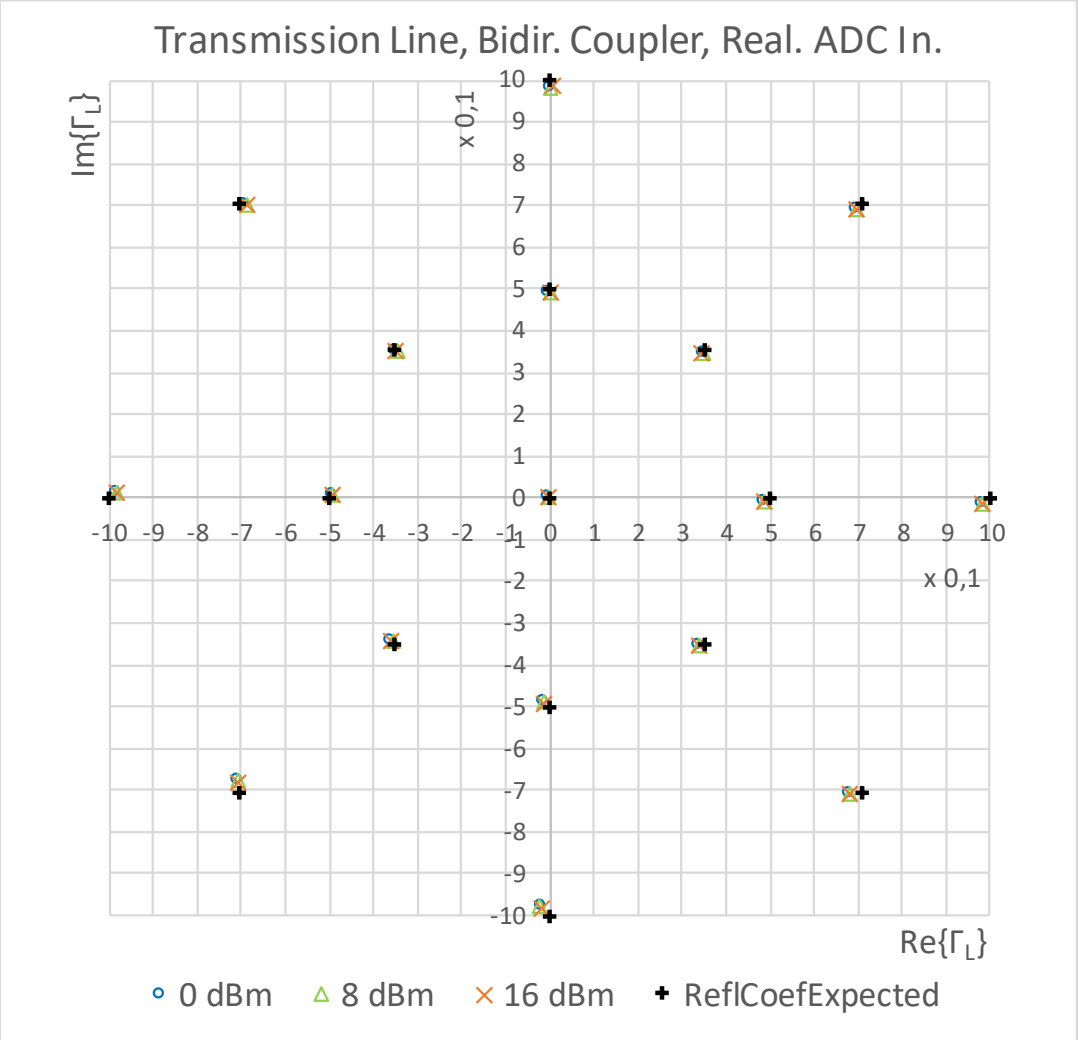

Figure 4.9: Simulation Results of Complete System using LTSpice

4.3.1 Statistics over LTSpice Simulation Results

Some statistics of the simulations done with LTSpice are displayed in Table 4.6. The given percentages are error magnitudes in the reflection coefficient measurement. The following abbreviations are used in the table:

P_{in} - Input power (Low, Medium, High)

MAG – Magnitude

PHA – Phase

ALL SIMP. – All submodels simplified, two signal sources instead of coupler

REAL. ADC INP. – Like before, but with realistic ADC input circuitry

R.ADC I. + CPLR – Like before, but now a coupler was used (still with two signal sources)

TML + CPLR + ADC – Like before, but now with transmission line and only one source.

It can be observed that again the input power does not make a big difference for the measurement error of the system, what can be expected, as the power acts on numerator and denominator of the reflection coefficient equation equally, thus the division cancels out this effect. But again, there is an influence of the reflection coefficient magnitude, even in the simulations without the directional coupler. This influence probably depends on the sampling effects in the ADC and on jitter in the sample clock as these affect the incident and the reflected signal equally. But as the incident signal does not change magnitude if the input power is constant, its noise contribution is also constant. However, the reflected signal changes magnitude according to the reflection coefficient magnitude (it is in the numerator of the reflection coefficient equation). It is still visible when using the directional coupler, but now there is also a dependency on the phase that could not be observed before (or was much weaker), so the phase dependency is really a coupler effect which is expected and must be resolved using a calibration method, for example short-open-load (SOL) as normally used in vector network analyzers (VNA) as a comparable (but more complex) device.

Generally, as expected, the errors get higher the more effects are included into the simulation, but with the coupler there was a comparably huge step up in the error magnitudes when compared with the system without the coupler, even though the coupler model itself was not including nonidealities.

Table 4.6: Statistics of System Simulations using LTSpice

Average over	ALL SIMP	REAL. ADC INP.	R.ADC I. + CPLR	TML + CPLR + ADC
ALL	0.47%	0.55%	1.14%	1.65%
LOW Pin	0.55%	0.58%	1.14%	1.68%
MED Pin	0.42%	0.56%	1.17%	1.68%
HIGH Pin	0.43%	0.51%	1.12%	1.61%
MAG 0.0	0.00%	0.00%	0.49%	0.49%
MAG 0.5	0.33%	0.39%	0.82%	1.18%
MAG 1.0	0.66%	0.78%	1.55%	2.28%
PHA 0°	0.56%	0.57%	1.26%	1.78%
PHA +45°	0.51%	0.58%	0.77%	1.56%
PHA +90°	0.42%	0.61%	0.63%	1.36%
PHA +135°	0.49%	0.58%	0.57%	1.29%
PHA 180°	0.56%	0.57%	1.10%	1.50%
PHA -135°	0.51%	0.58%	1.63%	1.99%
PHA -90°	0.42%	0.61%	1.86%	2.21%
PHA -45°	0.49%	0.57%	1.66%	2.12%
MAX. ERROR	0.83%	0.99%	2.51%	3.00%

4.3.2 Simulations with Random Noise

In a final simulation, random noise was included in the model. The noise amplitudes were adjusted to match the specifications of the ADC input channels, especially the Signal to Noise Ratio (SNR) in Nyquist sampling mode (about 72.7dB worst case, using the full power bandwidth limit of 575 MHz of the ADC). The SNR of the Nyquist sampling mode was used because this is kind of a reference value, for other sampling rates the SNR decreases or increases according to equation (23). This increase or decrease is already modeled through the sample-and-hold devices themselves.

Additionally, the output phase noise of the PLL (about 0.01° RMS from 10 kHz to 100 kHz) was modeled. The phase noise was added as voltage noise at the input of the PLL Voltage-Controlled Oscillator (VCO).

4.3.2.1 ADC Input Noise

With a maximum signal input amplitude V_{SIG} = 1V (sine signal, $V_{SIG} = V_i$ or V_r) and a worst-case Nyquist SNR of 72.7dB, the following RMS voltage noise amplitude can be obtained using (42) and (43):

$$V_{NADC} = V_{SIG,RMS} 10^{-\frac{SNR}{20}} \tag{42}$$

$$V_{NADC} = \frac{\sqrt{2}}{2} 10^{-\frac{72.7}{20}} \approx 0.16mV \tag{43}$$

4.3.2.2 PLL Phase Noise

Transfer function of VCO (44):

$$\frac{\theta_{VCO}}{V_C} = \frac{K_{VCO}}{s} \tag{44}$$

With the phase θ_{VCO} at the VCO output and the control voltage V_C.

The K_{VCO} parameter can be obtained from the data sheet of the Analog Devices ADF4360-9 integer-N PLL frequency synthesizer, or more easily from simulation with the Analog Devices Software ADIsimPLL (45). It must be multiplied by 2π to convert it to radians per second [52]:

$$K_{VCO} = 5.2752\pi \ \frac{Mrad}{V \cdot s} \tag{45}$$

The necessary voltage noise V_N at the VCO input for obtaining the phase noise θ_N is then (46):

$$V_N = \frac{s\theta_N}{K_{VCO}} \tag{46}$$

For steady state signals, the following expressions (47 and 48) hold:

$$s = j\omega \tag{47}$$

$$\omega = 2\pi f \tag{48}$$

For RMS values, the 90° phase implied by the j does not have any effect, so we can write (49):

$$V_N = \frac{2\pi f \theta_N}{K_{VCO}} \tag{49}$$

The center frequency of the PLL output [53] is $f = 7.99$ MHz.

ADIsimPLL also states that the circuit has a Root Mean Square (RMS) phase jitter of 0.01° from 10 kHz to 100 kHz. This equals in radians (50):

$$\theta_N = \frac{0.01°}{180°}\pi \tag{50}$$

This results in (51):

$$V_N = \frac{2 \cdot 3.14159^2 \cdot 7.99 \text{ MHz}}{2 \cdot 3.14159 \cdot 5.275 \text{ MHz}} \cdot \frac{0.01°}{180°} \text{V}_{\text{RMS}} \approx 0.26 \text{ mV}_{\text{RMS}} \tag{51}$$

4.3.2.3 Statistics over Random Noise Simulations

Due to a calculation error, about 200 times too much phase noise (equivalent to 53 mV_{RMS}) was added to the simulations. The frequency of the VCO output of 255.68 MHz was used instead of the PLL output of 7.99 MHz, what causes a factor of 32. Furthermore, K_{VCO} was given

in MHz/V and not in Mrad/s/V, introducing another factor of 2π. Thus, the results become very random, as shown in Table 4.7. This table is presented here to allow explanation of the investigation of the error sources.

Table 4.7: Statistics of Erroneous Random Noise Simulations using LTSpice

Average over	Random Noise Γ |Error|
ALL	37,13%
LOW Pin	39,68%
MED Pin	34,63%
HIGH Pin	37,06%
MAG 0.0	0,48%
MAG 0.5	23,94%
MAG 1.0	54,89%
PHA 0°	1,91%
PHA +45°	59,76%
PHA +90°	38,13%
PHA +135°	67,67%
PHA 180°	1,23%
PHA -135°	27,08%
PHA -90°	63,40%
PHA -45°	56,14%
MAX. ERROR	169,70%

PLL phase noise directly acts on the sample points of the ADC and is thus transferred to phase noise in the sampled signals, both reflected and incident. The phase of both signals then includes jitter but is still simultaneously sampled.

The error in the calculations was found by deducing that the correlation between reflection coefficient magnitude (MAG) and error magnitude is due to the higher power in reflected signal, i.e., a higher signal amplitude, that would lead to more noise in the sampled reflected signal if the PLL phase noise / clock jitter was the dominant error source (aperture error). And as the reflected signal is in the numerator of the reflection coefficient equation, this has direct influence on the reflection coefficient.

On the other hand, higher input power levels did not show the same effect, probably also due to the equation of the reflection coefficient calculation. As the phase noise acts on numerator and denominator equally (both are sampled simultaneously), the effect of different power levels cancels out, and the average error level takes over.

It was confirmed in a separate simulation run with just ADC SNR and no phase noise that the obtained errors are due to the phase noise or clock jitter (the contribution of the ADC SNR was minor).

Further investigations were carried out to confirm if the total amount of phase noise that was added was too high. Indeed, using the same random seed and time setup as before, but adding the noise directly as time delay at the output of the PLL (what is closest to reality),

there was only an extremely low influence of the realistic phase noise found. Unfortunately, the simulations with the time delay run much slower, so that one run takes in the order of one day and the complete simulation of 51 runs would probably take more than 50 days. Therefore, only the two worst case points (Mag.=0.5, Pha.=-90° and Mag.=1.0, Pha.=45°, both at P_{in}=20dBm) were simulated. The results were now 1.52% and 1.85% error.

4.4 Conclusion of Simulations

As the correct calculus for the phase noise was introduced into the document only after the simulations had been concluded, and as the real-world measurements in the PCB setup were already in preparation, the complete simulation run for the noise contributions was not executed anymore. However, the simulations were sufficient to show the validity of the approach and laid a good foundation for the following measurements and the understanding of the proposed system.

5 Design and Implementation

This section will focus on the practical implementation of the software defined radio frequency (RF) reflection coefficient measurement system based on under-sampling down-conversion, including electronics and software. Design considerations and implementation details will be described to offer a more complete view of the system.

5.1 Electronics

As the system is software defined, the electronics are considerably reduced. The directional coupler was moved outside of the designed printed circuit boards (PCBs) to allow for maximum flexibility and evaluating the influence of losses in this device more easily if required. This way, high quality bidirectional couplers can be tested as well as miniaturized more lossy ones including possibly integrated couplers. It only remains the analog-to-digital-converter (ADC) and its periphery, such as the phase locked loop (PLL) for sample frequency generation and the anti-alias-filter, as well as some digital bus drivers and input buffers at the digital interface to the microcontroller on which the software part of the system is running and that connects to the personal computer that records the data and offers a graphical user interface (GUI) for controlling the functions of the system.

5.1.1 Schematic

The schematic of the system PCB is shown in Figure 5.1 on the next page. For the schematic capture as well as for the PCB layout, the software Cadsoft Eagle 7.7 [54] was used. The schematic includes the circuity for the ADC (IC2), the pre-divider (IC10), the PLL (IC1), the sample clock driver (IC12) the digital interface (parallel bus clock gating IC3 and IC13 making use of the same logic family as the pre-divider and bus drivers IC4, IC5, IC6, IC7, IC8 and IC9 from another small-size, low voltage, fast logic family) and the voltage supply (IC11), all that is on the system PCB. The supply is based on the low noise low dropout voltage regulator LT1763 and several ferrites and capacitors for RF blocking. The complete system also includes an adapter PCB to connect the ribbon cable from the system PCB to the NUCLEO-F446RE microcontroller board [43]. The description of this board will be limited to showing the schematic (section 5.1.1.3), while the system PCB will be described in more detail.

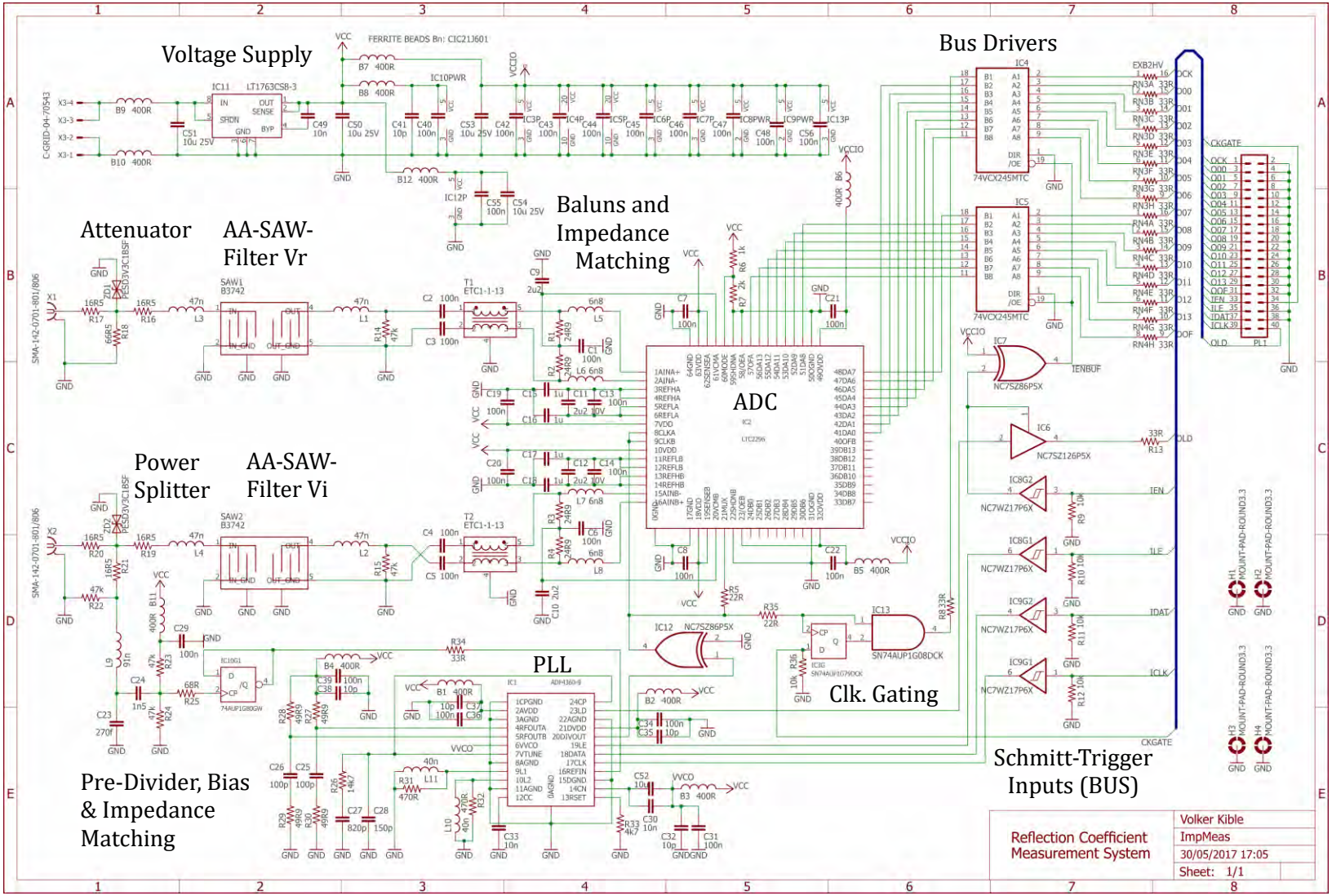

Figure 5.1: Schematic of Measurement System

5.1.1.1 Analog to Digital Converter (ADC)

The chosen ADC is the Linear Technology LTC2296, a 25 Msps 14 bit pipelined ADC that has a 575 MHz full power bandwidth sample and hold circuit and allows simultaneous sampling of two channels. The datasheet [39] already gives many recommendations for the schematic (in the applications information section), that were followed as closely as possible to avoid unexpected behavior.

5.1.1.1.1 Input Circuitry and Anti-Alias-Filters

The ADC requires a balanced input and some impedance matching for 400 MHz that is described in figure 8 on page 18 of the device datasheet [39]. Any application using a sampling ADC generally requires an analog anti-alias-filter (AAF) to limit the bandwidth of the sampled signal and prevent overlapping of the signal's aliases as described before. Often, a simple RC-lowpass is sufficient. However, as the application requires under-sampling, also unwanted lower frequency signal components must be suppressed. For the chosen under-sampling ratio and bandwidth, the according bandpass AAF must be narrow band and feature steep transitions between passband and stopband. Therefore, a 400 MHz surface acoustic wave (SAW) filter B3742 was chosen for the application. This filter is a ready-made component based on piezoelectric effect and electroacoustic coupling.

These SAW devices also require some (inductive) impedance matching at input and output (given in the data sheet of the SAW [55]), and as they are sensitive to electrostatic discharge (ESD) also some kind of ESD protection. Two bidirectional ESD protection diodes PESD3V3C1BSF [56] at the middle node of the power splitters / attenuators were employed. This location promises the minimum effect on the signal amplitude and phase difference while at the same time protecting the ESD-Diodes themselves from unallowably high input power. The splitters / attenuators reduce the input signals to amplitudes allowable for the following components and direct (split) the incident signal into two paths, one to the AAF of the incident ADC channel, the other to the impedance matching circuit of the pre-divider for the PLL.

5.1.1.1.2 Mode Selection

As described in the datasheet of the LTC2296 ADC, the mode pin controls the output format and the use of the clock duty cycle stabilizer. For setting the latter to active, the pin must be either at 1/3 Vcc or at 2/3 Vcc for offset binary and 2's complement output format, respectively. In the schematic, the last option is displayed, but it can easily be changed to any other mode just by adjusting the resistor values while populating the board.

5.1.1.1.3 Parallel Interface

The MUX-pin is connected to the sample clock signal, so that both data outputs are multiplexed on one parallel data bus with 15 lines in such way that the data from channel A is output during the clock high phase and channel B during clock low phase. The bus consists of the 14 bit of the ADC sample and an additional signal for overflow of the channel. This way, only half the number of pins of the microcontroller are needed, but the effective data clock is doubled. As the ADC is sensitive to loading of the output pins, additional digital bus drivers 74VCX245 [57] are included. These also protect the expensive ADC from damage due to bus faults.

5.1.1.2 Phase-Locked Loop (PLL)

The integer-N-PLL with integrated voltage-controlled oscillator (VCO) Analog Devices ADF4360-9 [38] is included in the system for generating the required under-sampling sample frequency with a fixed ratio compared to the input signal. For this purpose, the PLL reference signal is derived directly from the incident input signal (that should normally have a higher amplitude than the reflected one).

5.1.1.2.1 Pre-Divider

As the datasheet states a maximum frequency of 250 MHz for the reference signal input, a frequency pre-divider had to be applied. The D-type flipflop 74AUP1G80 [37] with inverted output fulfills this purpose.

5.1.1.2.2 Voltage Controlled Oscillator (VCO)

The VCO makes use of two external size 0402 inductors that should be mounted in a 90° angle orientation towards each other to reduce parasitic coupling. Parallel to each of these inductors, a resistor of 470 Ohm is required. For the VCO frequency band of interest, the inductors should have 40 or 41 nH. Additionally, the VCO has an output that must be impedance matched even if it is not used to prevent erratic behavior. For this reason, four 50 Ohm resistors and two 100 pF capacitors are applied. For the application, the used frequency output is the DIVOUT output that allows an additional division of the VCO frequency and, in A/2 mode, 50% duty cycle.

5.1.1.2.3 Loop Filter

The loop filter was designed using Analog Devices ADIsimPLL [51]. The simplest topology was used (that is also the one adding less phase noise), resulting in a setup of two capacitors (820 pF and 150 pF) and one resistor of 14.7k. The use of the standard design equations for these components yielded similar results.

5.1.1.2.4 RF Blocking (Ferrites and Capacitors)

As demonstrated in the datasheet of the ADF4360-9, it is important to prevent RF output from one part of the circuit from interfering with other parts of the circuit. For this purpose, blocking ferrite beads CIC21J601 and capacitors (various values) were employed on all important power supply pins.

5.1.1.2.5 Serial Peripheral Interface (SPI)

The ADF4360-9 works with a serial interface which is compatible to SPI. The three latches “R Counter Latch”, “Control Latch” and “N Counter Latch” (each 24 bits) must be written in exactly this sequence with at least 15 ms pause between the “Control Latch” and the “N Counter Latch”. The allocation of the bits of all latches can be found in the data sheet. The device does not have a data output or any other means of knowing if the programming worked. The only information that it offers is a lock detect output that shows when the PLL is in lock. For protecting the inputs of the PLL, Schmitt-trigger input buffers of the type NC7WZ17 [58] were used at the microcontroller interface. This is not required by the datasheet, but as the ADF4360-9 is a more costly and harder to solder component than the buffers, and as the microcontroller here is not directly connected (it attaches to the system board via a ribbon cable as stated before), this was the more robust solution.

5.1.1.3 Interface to NUCLEO-F446RE

The interface to the microcontroller board NUCLEO-F446RE [43] is a ribbon cable including all data lines from and to the system board.

On the system PCB side, there is a connector for the ribbon cable and digital bus drivers and Schmitt-trigger input buffers for all signals.

Additionally, there exists an adapter board to connect the ribbon cable to the NUCLEO.

The adapter board to connect the ribbon cable to the NUCLEO-F446RE has the schematic shown in Figure 5.2.

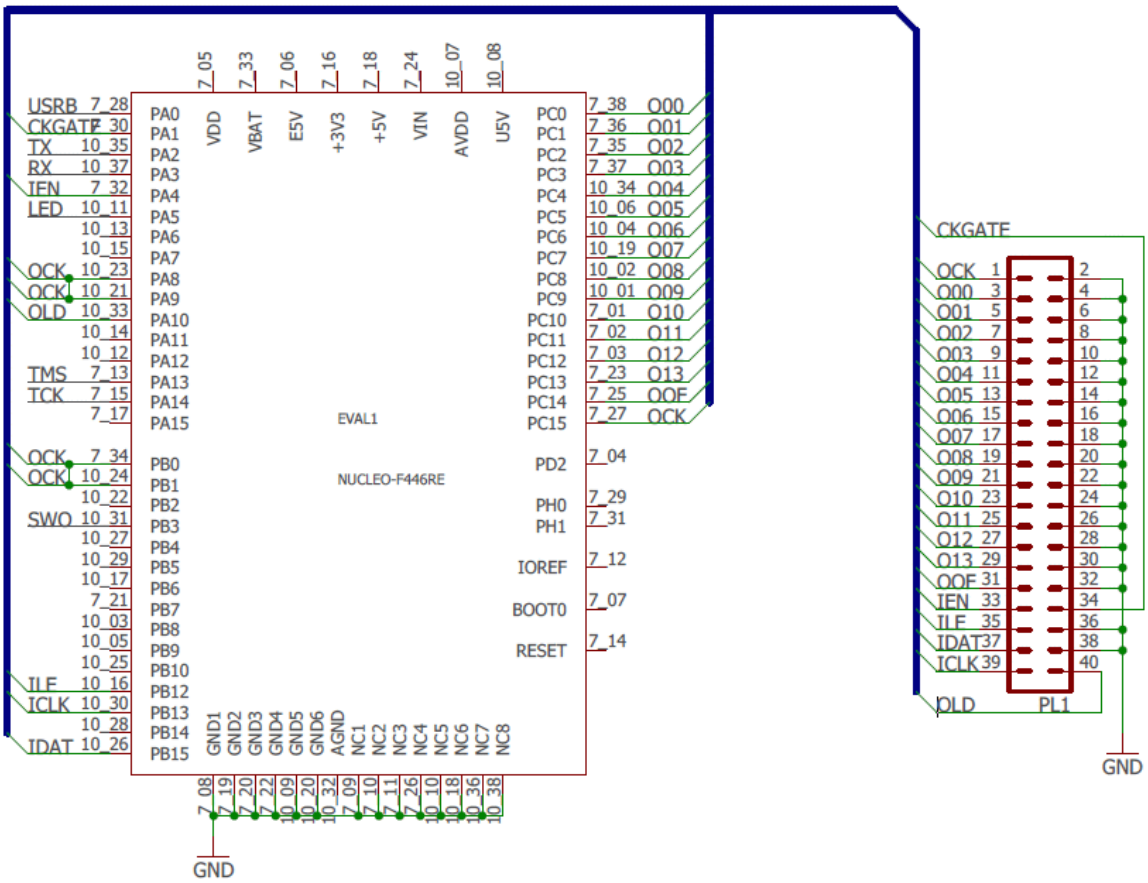

Figure 5.2: Schematic of NUCLEO-Adapter

5.1.1.3.1 Lines of Ribbon Cable Connector

The interface consists of the following lines:

- IEN: Input "Enable Outputs" for whole interface,
- ILE: Input "Load Enable" for PLL,
- IDAT: Input "Serial Data" for PLL,
- ICLK: Input "Serial Clock" for PLL,
- OOF: Output "Channel Overflow" of ADC,
- O13: Output "Bit 13" of ADC,
- O12: Output "Bit 12" of ADC,
- O11: Output "Bit 11" of ADC,
- O10: Output "Bit 10" of ADC,
- O09: Output "Bit 9" of ADC,
- O08: Output "Bit 8" of ADC,
- O07: Output "Bit 7" of ADC,
- O06: Output "Bit 6" of ADC,
- O05: Output "Bit 5" of ADC,
- O04: Output "Bit 4" of ADC,
- O03: Output "Bit 3" of ADC,
- O02: Output "Bit 2" of ADC,
- O01: Output "Bit 1" of ADC,
- O00: Output "Bit 0" of ADC,
- OCK: Output "Parallel Clock" for ADC data,
- OLD: Output "Lock Detect" of PLL,
- CKGATE: Input "Clock Gating" for activating the "Parallel Clock" separately,
- Various ground connections, used as a shield between the highspeed data lines.

5.1.1.3.2 Digital Bus Drivers and Schmitt-Trigger-Input Buffers

Electronically, the interface consists of digital bus drivers (2x 74VCX245MTC [57] and 1x NC7SZ126P5X [59]) as well as Schmitt trigger input buffers (2x NC7WZ17P6X [58]). The circuit for clock gating (see below) also has a Schmitt-trigger input. These digital circuits make the board less sensitive to bus errors and electromagnetic interference. The outputs of the board are normally off, only when a high signal is applied to the IEN-pin, they are enabled.

5.1.1.3.3 Clock Gating Circuit

An additional high signal at the CKGATE-pin is required to enable the parallel data clock. This ensures that the microcontroller uses the rising edge first to read the first data. To synchronize the CKGATE signal with the parallel data clock, a SN74AUP1G79DCK D-type flipflop with a positive output [60] is used together with an AND gate of the type SN74AUP1G08DCK [61]. The flipflop has Schmitt-trigger inputs.

5.1.2 Printed Circuit Board Layout

The system PCB is a mixed signal PCB with an analog RF part and a digital part. As such, it requires some special precautions for the layout (shown in Figure 5.4 to Figure 5.7). Unluckily, due to the high costs of such a technology, it was not possible to manufacture the

board as impedance-controlled RF board. So, the board was designed to work with standard PCB manufacturing technology, even though that is far from ideal from the RF point of view. The ready manufactured PCB (size 90 mm x 72 mm) is shown in the following Figure 5.3.

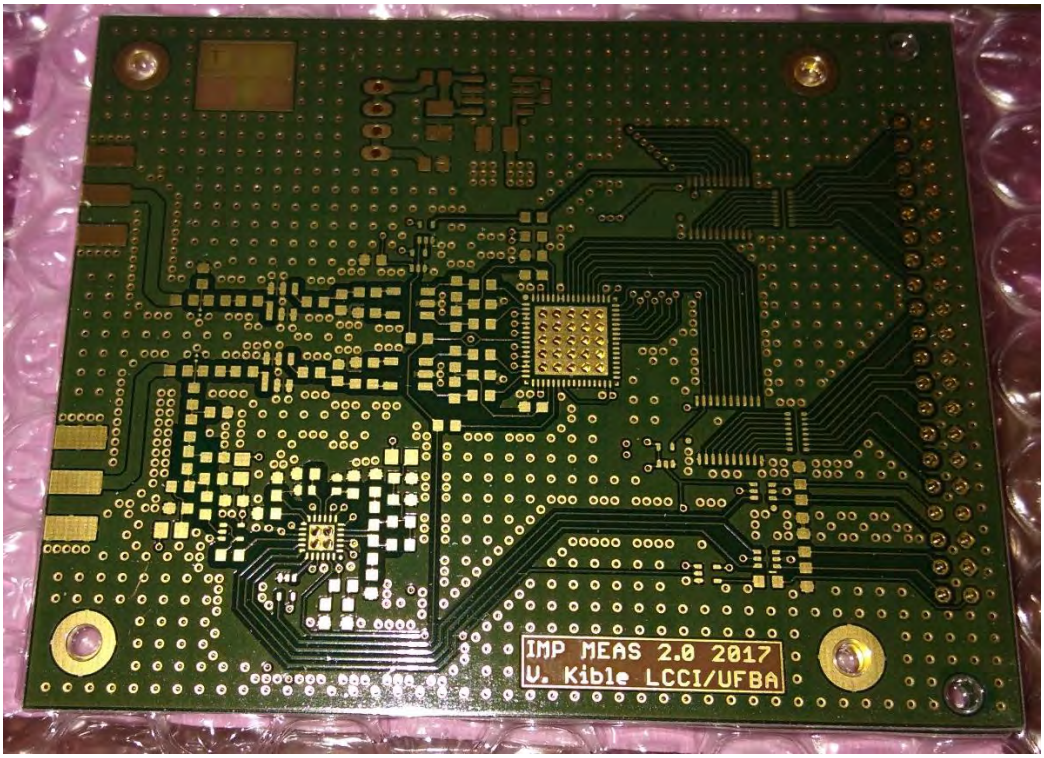

Figure 5.3: Photo of System PCB

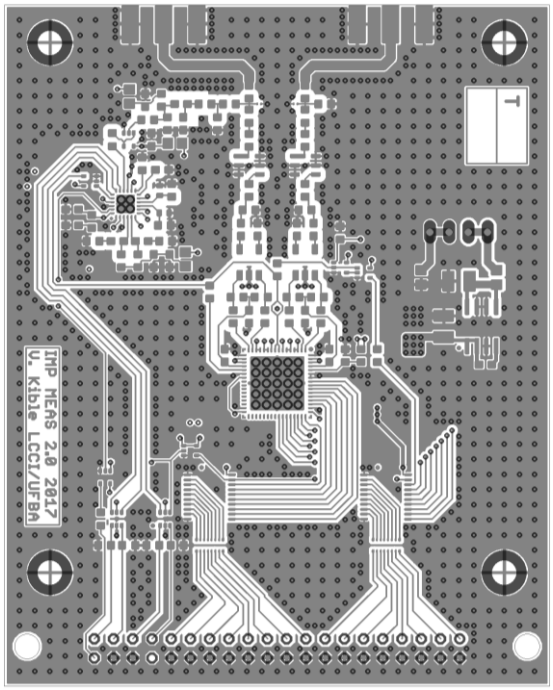

Figure 5.4: Layout top layer (TOP)

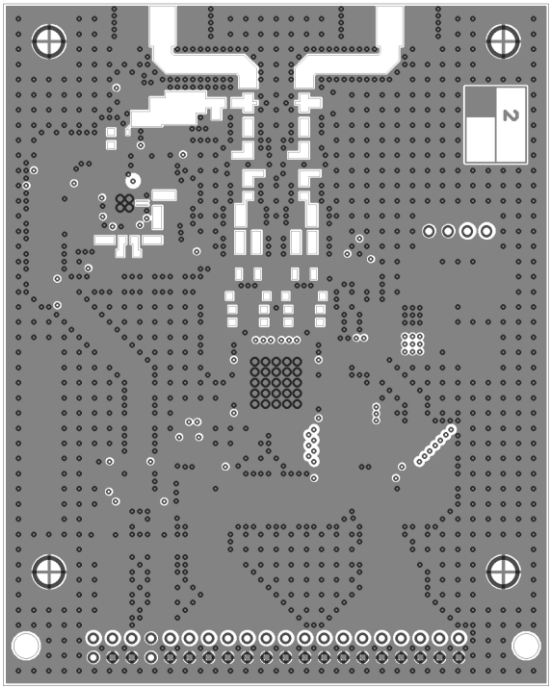

Figure 5.5: Layout upper inner layer (L2)

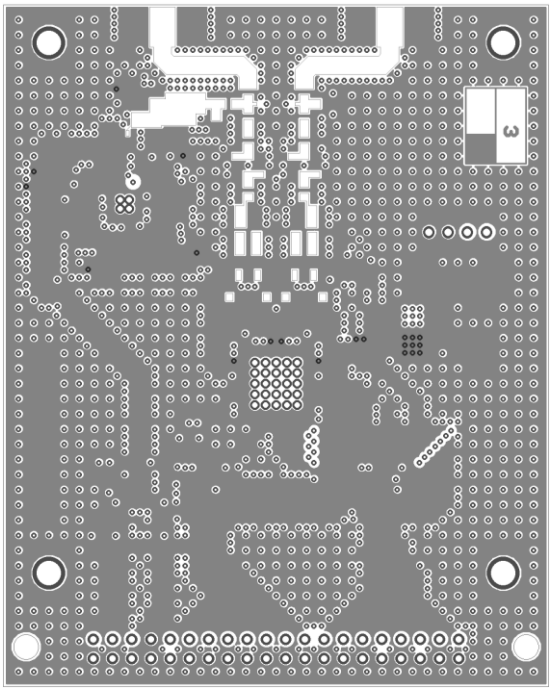

Figure 5.6: Layout lower inner layer (L3)

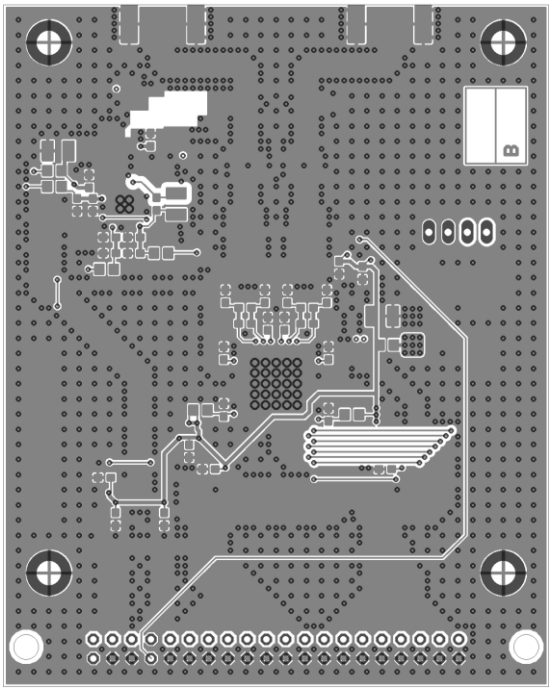

Figure 5.7: Layout bottom layer (BOT)

5.1.2.1 RF Part

The RF part (on the upper part of Figure 5.4 to Figure 5.7) guides frequencies in the very-high frequency (VHF – at the PLL) and ultra-high frequency (UHF) range. As such, impedance-matching and reduction of unwanted parasitic effects is crucial.

5.1.2.1.1 Symmetry

To minimize phase shift between the two input channels (incident and reflected), the layout of these two channels must be as symmetrical as possible. The layout guarantees symmetry down to the manufacturing tolerances of board and soldering, what should be sufficient. The symmetry is only broken by the fact that the splitter in the incident channel also supplies the pre-divider of the PLL. The input of the pre-divider is impedance matched to the incident channel and thus should act on the UHF signals the same way as the attenuator on the reflected signal branch.

5.1.2.1.2 Solid Ground Plane

To obtain the best possible RF ground, a solid ground plane was applied on one of the inner layers. Additionally, all unused space in the outer layers of the board is filled with a ground plane as well. The inner ground plane only has some small holes to allow a reduction of parasitic capacitance under the components and lines (shown in Figure 5.5), but most of these holes are filled by the outer ground plane on the back side (shown in Figure 5.7) that are connected by many vias (metallized through holes in the PCB). An exception is the void

below the input impedance matching circuitry of the pre-divider for the PLL to reduce parasitic capacitance.

5.1.2.1.3 Reduction of Parasitic Capacitance

To reduce the parasitic capacitance, the supply planes (ground and Vcc, Figure 5.5 to Figure 5.7) have holes under critical traces and component pads, especially in the main RF signal path of V_i and V_r and under the impedance matching of the pre-divider. This way only the stray capacitance to the border of the hole plays a role for the most critical area, and at other places the distance to the ground supply plane is made much wider.

5.1.2.1.4 Small Size Tracks and Components

To reduce parasitic transmission line effects, capacitances and inductances, the tracks between the components are kept as short as possible, and small size components are used. However most passive components were chosen to be size 0603 to allow easier handling. Only the inductors of the PLL VCO are 0402 and the ESD protection diodes 0201 (these are not the only parts on the board that will be difficult to solder). Decoupling capacitors and blocking ferrite beads are mounted close to the supply pins of the integrated circuits (ICs).

5.1.2.2 Digital Part

The digital part is generally much less sensitive to unfortunate PCB layout; nevertheless, the decoupling capacitors (and some ferrite beads) are mounted close to the supply pins of the ICs, just like in the RF part. However, there are some more considerations:

5.1.2.2.1 Tracks

As digital signals typically have sharp edges that contain much higher frequency content than the signal frequency, it is good practice to keep a minimum distance between digital tracks that is bigger than the distance required by the PCB design rules. This reduces capacitive coupling. Finally, ringing of the signals on the tracks can be reduced by small series resistors (33 Ohm or the like) that serve as a kind of impedance matching.

5.1.2.2.2 Four Layer PCB

For routing the digital (and the RF) signals, it is of advantage to use two routing layers, so the total size of the sub-circuits and the board can be reduced. Additionally, the supply planes under the tracks reduce inductive coupling and allow shorter tracks to the decoupling capacitors.

5.1.3 Populated Board

The completely populated system PCB with marked functional blocks is shown in Figure 5.8. Please also note the ferrite core on the bus cable to the microcontroller-board, that should prevent cable bound RF noise from the PC and the microcontroller to enter the system PCB. For the USB-cable from the microcontroller board to the PC, a type with included ferrite cores was chosen for the same reason as well.

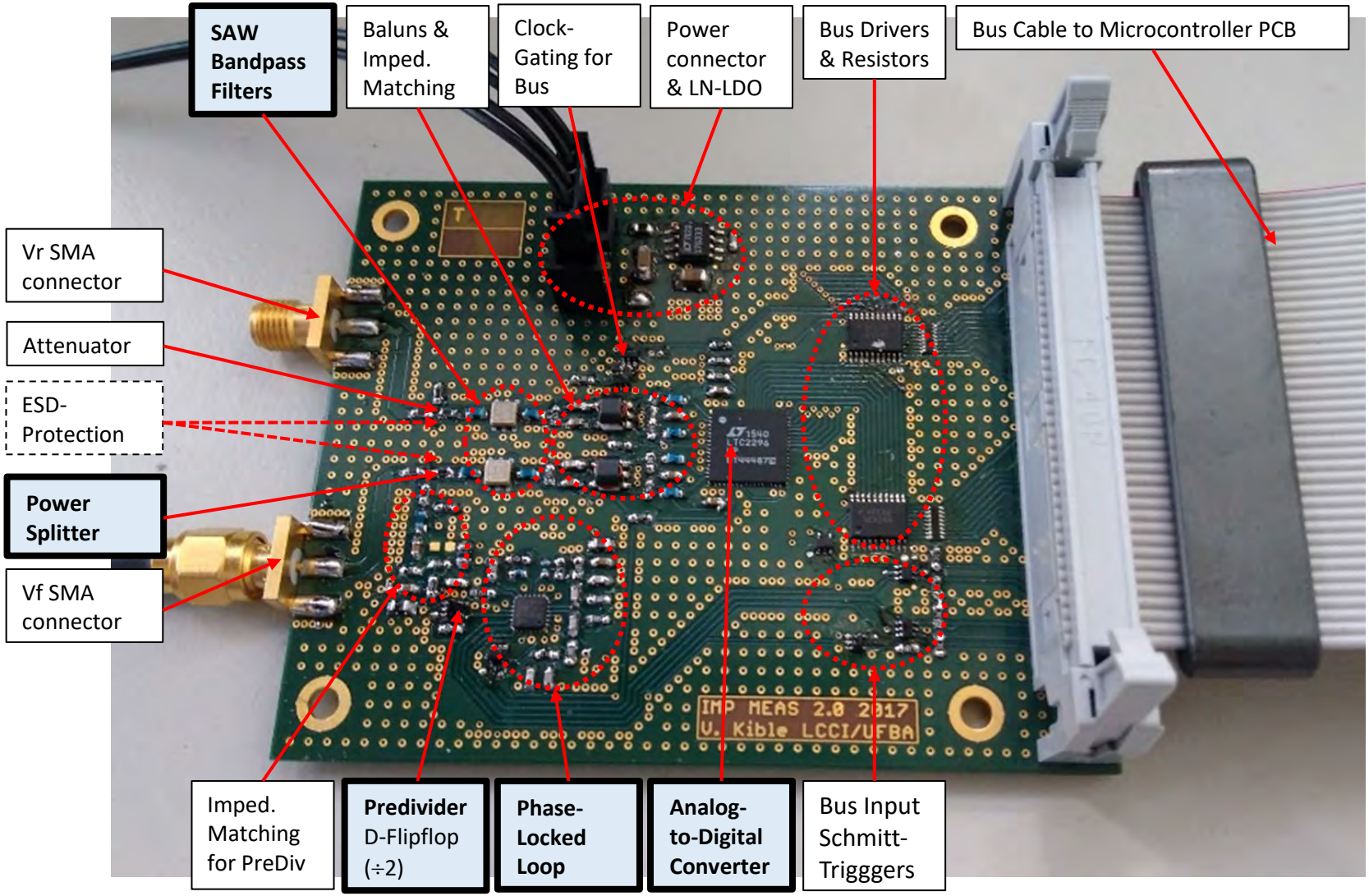

Figure 5.8: Populated System PCB

5.1.4 Modification of the Microcontroller Board

Some minor modifications on the NUCLEO-F446RE microcontroller board are necessary for the project. Most of these modifications take place on the back (solder side) of the PCB, if the modification is on the front side, it is stated separately. The board is already prepared for being modified, there are solder bridges for many pins of the STM32F446RE microcontroller that are linked to other functions of the PCB. Only a solder iron is required to do these modifications, except for 5.1.4.3 that requires a small, isolated wire. For reference, the layout of the NUCLEO 64 board [43] is shown in Figure 5.9 and Figure 5.10.

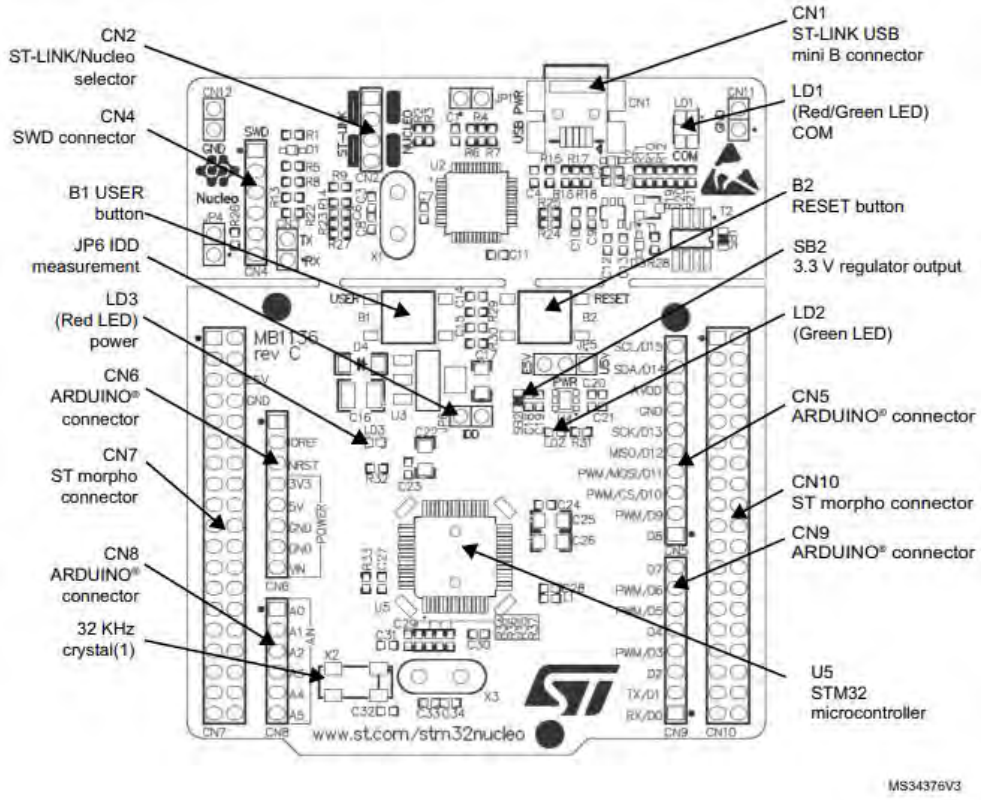

Figure 5.9: Nucleo 64 Top Layout [43]

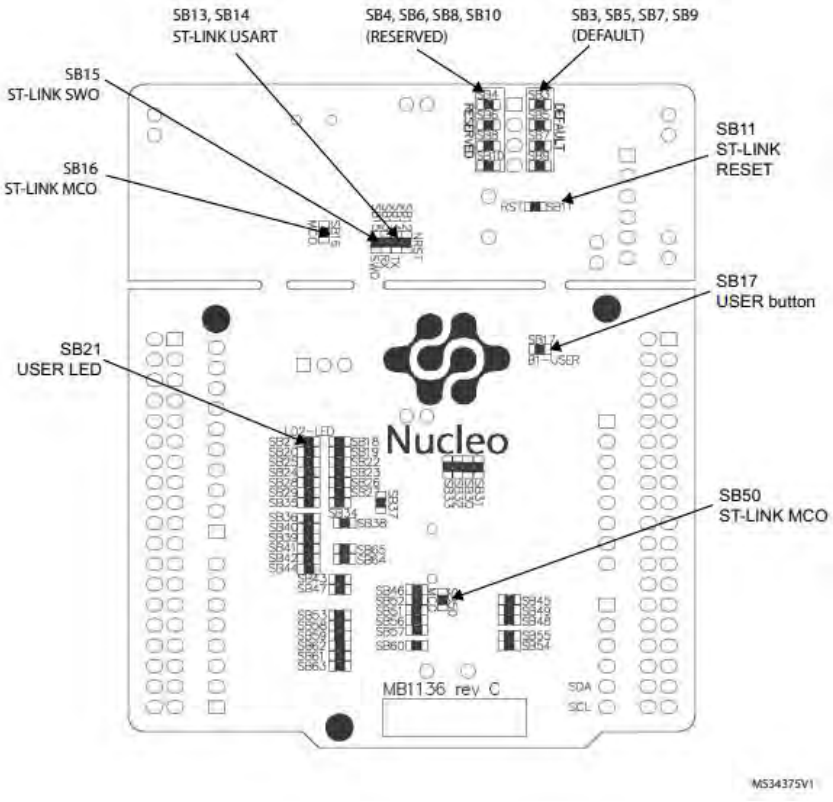

Figure 5.10: Nucleo 64 Bottom Layout [43]

5.1.4.1 Removing Bridges

SB17 (B1-USER) is disconnected to free the microcontroller pin PC13 for parallel data input. Also see 5.1.4.3. R34 and R36 on the front side of the PCB are removed as well to disconnect the X2 crystal to free PC14 and PC15, also for parallel data input.

5.1.4.2 New Bridges

Two new bridges are connected at the locations SB48 and SB49 (X2 crystal) to connect PC14 and PC15 to the external connector CN7 for parallel data input. This means that R34 and R36 must be removed.

5.1.4.3 Jumper Cable

To connect the on-board blue user button to the microcontroller pin PA0, a jumper cable is connected between the opened bridge SB17 at the button-side and the backside of the Arduino-connector CN8 pin 1 (A0).

5.2 Software

As the system is software defined, the software development plays a key role in the system design process. The software consists of two parts that run on two different processors, the slave microcontroller software that runs on the STM32F446RE [40], [41] on the NUCLEO-F446RE [43] board and that implements the software-defined part of the RF reflection coefficient measurement system, and the master personal computer (PC) software that offers

a graphical user interface (GUI) for controlling the system and the possibility to record and save the measured data for further processing (for example using MATLAB or Microsoft Excel or the like).

5.2.1 Microcontroller Software

The microcontroller software implements the reflection coefficient measurement from the two under-sampling down-converted wave forms (incident end reflected signal) using the Cortex M Software Interface Standard (CMSIS) real fast Fourier transform (rFFT) [45] and simple division of complex numbers. It also configures the PLL IC according to the needs of the project and offers various means of communication, such as serial peripheral interface (SPI) for configuring the PLL, 16 bit parallel data reception from the ADC, and universal asynchronous receiver transmitter (UART) to connect to the PC via the ST-LINK/V2-1 on-board debugger on the NUCLEO-F446RE [43] that serves as a bridge to the universal serial bus (USB) virtual COM (VCOM) port functionality. For using the hardware (internal and external peripherals) efficiently, various drivers were programmed. The total size of the output *.elf-file of the program is 417 kB. The main routine and the PLL driver together have less than 60 kB. All microcontroller programming was executed in a professional integrated development environment (IDE).

5.2.1.1 Integrated Development Environment (IDE)

As for any programming task, an IDE including an editor for the language of choice, a compiler, a linker, and a debugger for the target of choice is required to program the STM32F446RE microcontroller [40], [41]. Considerations concerning this IDE are collected in this section.

5.2.1.1.1 Atollic TrueSTUDIO for ARM 7.1.2

The IDE of choice for the programming task at hand is the Atollic TrueSTUDIO because it offers vast support for the NUCLEO-F446RE [43] board (and many other development boards and microcontrollers from the ARM family). It is based on the Eclipse IDE, but already comes with build- and startup files and pre-configurations for the target. It also allows the use of Eclipse plugins. It also offers support for the hardware abstraction layer (HAL) microcontroller peripheral drivers from ST Microelectronics [62] that are required when working with the ST Microelectronics STM32CubeMX code generator [47].

The Atollic TrueSTUDIO IDE is available for free download after registration at [46]. Documentation is also available there, including how to install the STM32CubeMX Eclipse plugin [63] in TrueSTUDIO. The code generated by STM32CubeMX can directly be used in TrueSTUDIO. A view of the C language code editor of TrueSTUDIO is shown in Figure 5.11.

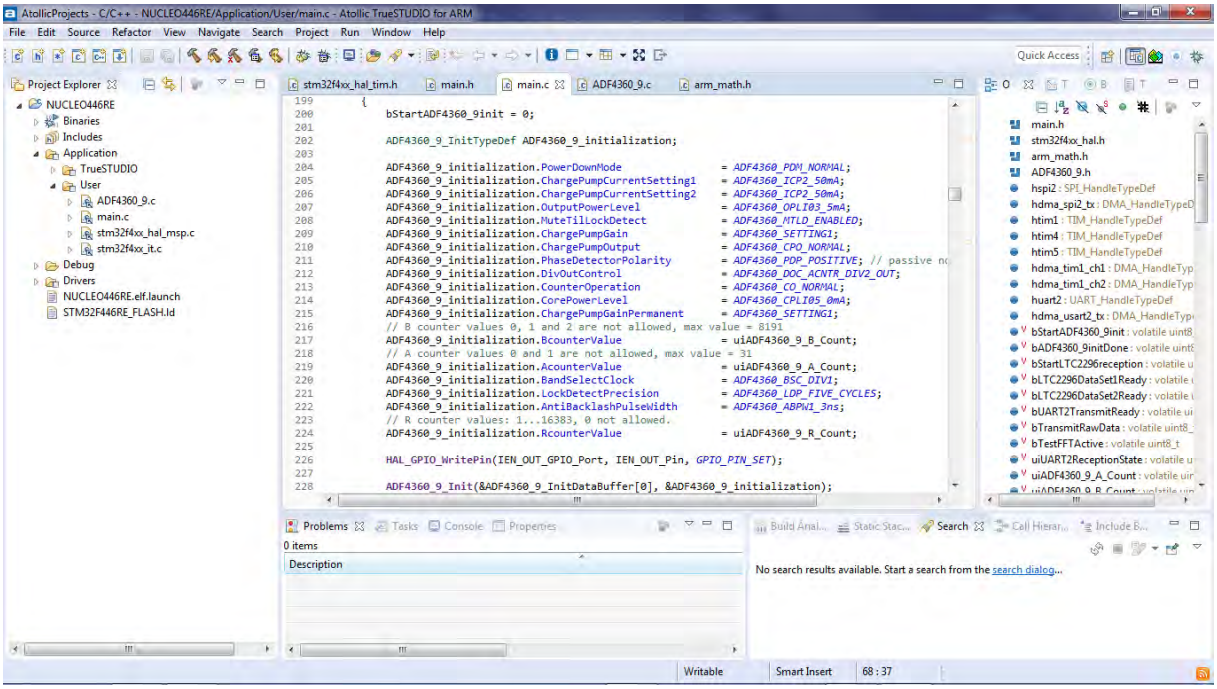

Figure 5.11: Atollic TrueSTUDIO Code Editor

5.2.1.1.2 ST Microelectronics STM32CubeMX

The ST Microelectronics STM32CubeMX code generator Eclipse plugin [63] integrates seamlessly into Atollic TrueSTUDIO. With this tool, the assignment of pins and direct memory access (DMA) controller channels to other peripherals is easy to do. Just as simple is the configuration of the otherwise complex clock distribution system of the device.

The pin assignment view of STM32CubeMX is shown in Figure 5.12. However, there are many other views to allow for configuring all functions of the device. It is even possible to estimate the power consumption.

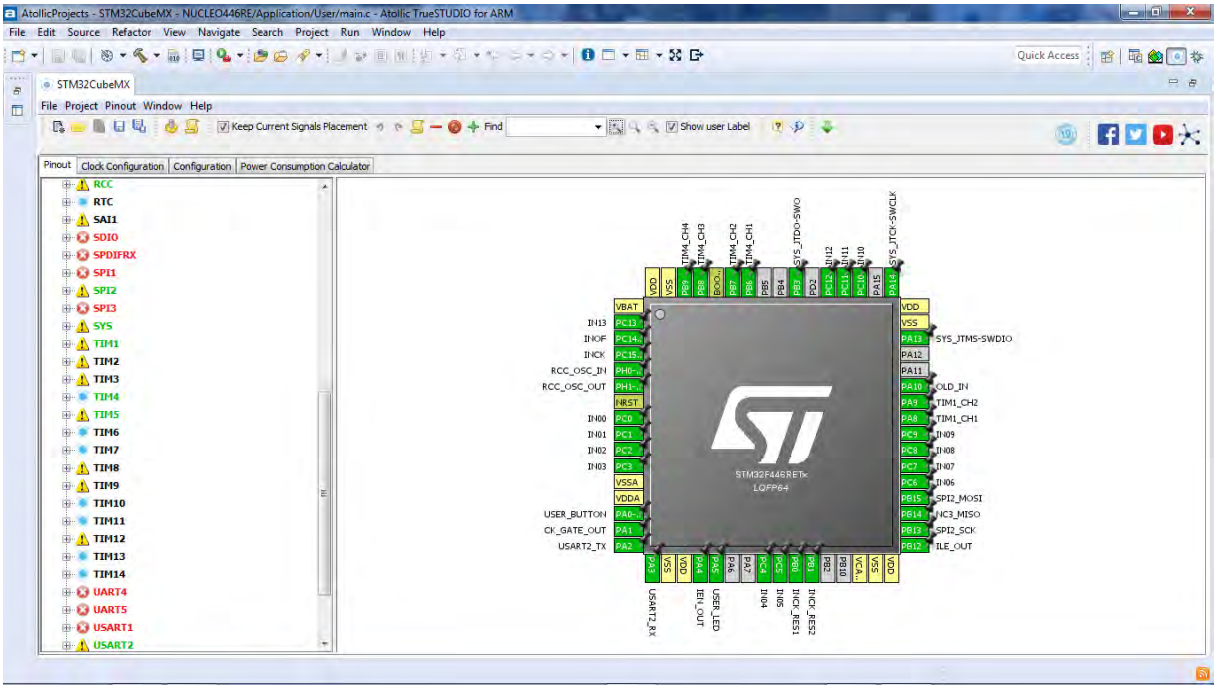

Figure 5.12: STM32CubeMX Pin Assignment View

The generated initialization code and drivers are based on the new HAL library of ST Microelectronics. As the HAL library [62] is different from the old standard peripheral library (SPL), it was necessary to learn how to use this library correctly to fine-adjust the drivers or add functionality by hand. However, the old and the new library also have many things in common, what made the changes easier.

5.2.1.2 Peripheral Drivers

With the help of STM32CubeMX and the HAL library, the necessary drivers for the internal and external peripherals of the microcontroller were written. Code from the programs for application note AN4666 "Parallel synchronous transmission using GPIO and DMA" [42] and from a driver supplied by Analog Devices for the ADF4360 PLL family [64] was also used.

5.2.1.2.1 Drivers for Internal Peripherals of the Microcontroller (HAL)

The STM32CubeMX helps a lot with generating the code for controlling the internal peripherals of the device, but the code defining the use and final behavior still must be written by hand as stated below. Code for the following peripherals was generated:

- Reset and Clock Control (RCC) for input of 8 MHz from the ST-LINK/V2-1 as high-speed external (HSE) and generating 180 MHz core frequency and various peripheral clocks from it.
- Nested Vectored Interrupt Controller (NVIC) for serving the interrupts of the other peripherals in the following priority order: (1) DMA2-Stream1, (1) DMA2-Stream2, (2) USART2, (3) SPI2, (4) TIM5, (5) DMA1-Stream4, (6) DMA1-Stream6, (12) EXTI-Line0. This also creates the interrupt handler routines. For user code, the according HAL callback must be overloaded (or defined).
- External Interrupt (EXTI) for receiving the signal of the user button.
- Direct Memory Access Controller 1 (DMA1) Stream 4 for SPI2_TX (transmit) DMA transfer request (interface to PLL) and Stream 6 for USART2_TX (transmit) DMA transfer request (interface to PC).
- Direct Memory Access Controller 2 (DMA2) Stream 1 and Stream 2 for TIM1_CH1 and TIM1_CH2 transfer request, respectively, for parallel data reception from the ADC.
- General Purpose Input Output (GPIO) for all necessary configuration of input and output pins of the microcontroller according to the interface description in section 5.1.1.3. For switching software-controlled pins on and off, the according HAL routine "HAL_GPIO_WritePin" must be used.
- Serial Peripheral Interface 2 (SPI2) for transmission only (of the register data for the PLL): frame format Motorola, data size 8 bits, most significant bit first, prescaler 32 for 1.4 Mbit/s baud rate, clock polarity low, clock phase first edge, cyclic redundancy check calculation disabled, slave select per software (used for load enable). The DMA request was configured to transfer single bytes from memory to the peripheral incrementing the memory address in normal mode. In the code, the HAL transmission via DMA routine "HAL_SPI_Transmit_DMA" must be added by hand, and the order of the bytes to transmit must be modified as needed.
- TIM1: 16 bit Advanced-Control Timer 1 for receiving the synchronous data clock for the parallel data reception from the ADC and accordingly triggering the DMA2 Stream 1 and Stream 2 request (TIM1_CH1 and TIM1_CH2 input capture on rising and falling

edge of the clock, respectively). The DMA requests were configured to transfer half words from the GPIOC peripheral to memory and auto increment the memory address in normal mode with very high and high priority, respectively. An own routine "TIM1StartDMA" to start the parallel synchronous data transmission correctly based on [42] and callback routines was written.

- TIM4: 16 bit General Purpose Timer 4 is only used for debug means and was configured to output the same PWM signal on four channels (PWM Generation Channel 1…4 mode 1 and pulse 5, fast mode disable and polarity high). The prescaler was set to 0 and the counter period to 9, resulting in a 50% duty cycle and 9 MHz frequency, for simulating the synchronous parallel data clock. The outputs must be enabled by hand in the code using "TIM_CCxChannelCmd".
- TIM5: 32 bit General Purpose Timer 5 was used as time base in the PLL driver. As such, no IC/OC channels were used, just the basic timer in down-counting mode with a prescaler of 22 and a timer period of 61999 (changed in software by hand using the proprietary function "TIM_Base_Set_Count" based on HAL functions) The timer is enabled using "__HAL_TIM_ENABLE" when needed and disabled using "__HAL_TIM_DISABLE" in the interrupt callback "HAL_TIM_PeriodElapsedCallback".
- USART2: Universal Synchronous/Asynchronous Receiver Transmitter 2 was used in asynchronous receive and transmit mode, 115200 bit/s baud rate, 8 bits word length, no parity and one stop bit. The DMA request USART2_TX was used to transfer single bytes in normal mode with memory address incrementing from the memory to the peripheral. For transmission, the HAL routine "HAL_UART_Transmit_DMA" was used. For reception, an interrupt-based reception scheme using "HAL_UART_Receive_IT" was adopted as only few bytes are received at once and speed is uncritical. The UART reception complete callback "HAL_UART_RxCpltCallback" was overwritten to implement the communication protocol with the PC.

5.2.1.2.2 Driver for PLL

For adapting the PLL, a driver was written (files ADF4360_9.c and -.h). A struct holding all the possible configurations for the device was assembled and enumerations and defines were used for the values to be written into the fields of the struct, making the code readable and easy to debug. The driver is used by first writing the desired values into the struct, then passing it to "ADF4360_9_Init" to initialize the registers that will be transmitted to the device and then calling "ADF4360_9_ WriteAllDeviceRegisters" that follows the protocol and requirements from the ADF4360-9 datasheet to write the registers into the device latches using SPI2 and TIM5. "ADF4360_9_Init" does basic plausibility checking and correction. The timer interrupt callback "HAL_TIM_PeriodElapsedCallback" must also set the "Load Enable" pin to low and increment the variable "ADF4360_9_WriteStepNew". This keeps the driver construction simple and does not block the microcontroller during the process.

5.2.1.2.3 Driver for ADC

For adapting the ADC, only a parallel synchronous data reception routine like in [42] had to be implemented. The complete GPIOC was used as data source and two DMA transfers were configured for the rising and the falling edge of the data clock, each transferring 2x32 single 16bit words sequentially and intermittent to two different arrays that can later be passed to the rFFT algorithm. In the function "TIM1StartDMA", first the four necessary

callbacks are defined, then the DMA is started with interrupt activated using "HAL_DMA_Start_IT" for both channels, then the timer channel DMA request is enabled on both channels using "__HAL_TIM_ENABLE_DMA", then the timer input capture channels are enabled using "TIM_CCxChannelCmd", and finally the clock gating pin is set high using "HAL_GPIO_WritePin". When the DMA transfer is complete (after 32 values), the callbacks are called. In the callbacks, the flags for the main program are set (the main program needs both flags set to proceed), then the according capture compare channels are disabled using "TIM_CCxChannelCmd" and the respective DMA transfer request of the timer capture compare channel is disabled using "__HAL_TIM_DISABLE_DMA". The main program sets the clock gating pin low again.

5.2.1.3 Main Program

The main program calls all configuration routines for the various peripherals (some of which were generated by STM32CubeMX) and initializes the rFFT data structures using "arm_rfft_fast_init_f32". Before entering the central infinite loop, the communication with the PC is started by receiving one byte using "HAL_UART_Receive_IT". In the infinite loop, the different flags are checked and the according part of the program is executed, always ensuring short response times and cyclic execution. Flags are set in the PC and PLL interface and in the synchronous parallel data reception callbacks and reset in the according part of the main loop right after they were positively checked, ensuring that the code is always executed only once. A data flow diagram of the reflectometer microcontroller software main routine is shown in Figure 5.13:

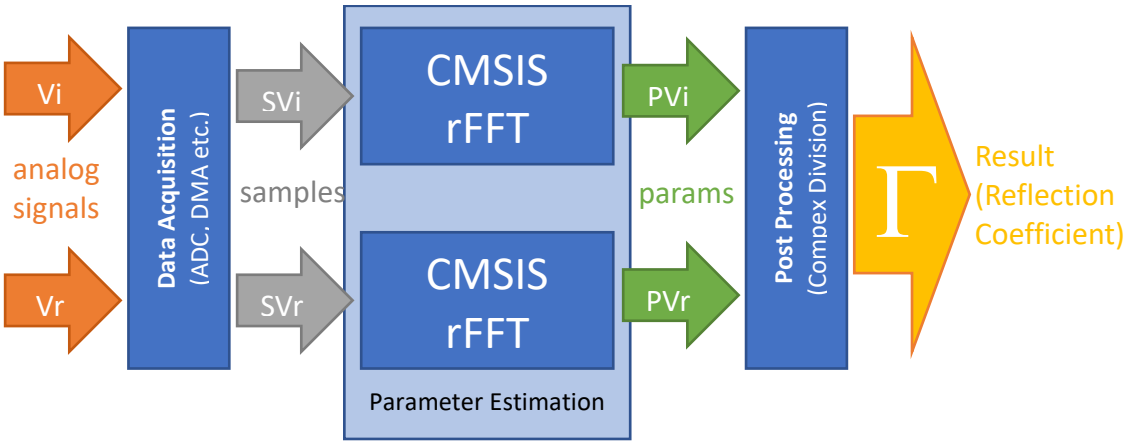

Figure 5.13: Data Flow Diagram of Microcontroller Software

5.2.1.3.1 CMSIS arm_rfft_fast_f32

The Cortex M Software Interface Standard (CMSIS) arm_rfft_fast_f32 function is a so-called real fast fourier transform (or inverse transform). In the direction time domain to frequency domain (ifftFlag = 0), it expects a real valued array with the time data and outputs a complex valued array with complex amplitudes, both arrays have the same length (here 32). As the DC component and the component at the Nyquist frequency ($f_s/2$) contain no phase information, they are packed at the array positions 0 and 1, respectively. These two array positions form FFT bin 1 together. All other positions are interleaved real and imaginary (the even array positions 2, 4, 6,… contain the real values and the odd positions 3, 5, 7,… the imaginary

values). Bin 2 is then found in array positions 2 and 3, and bin 3 is found at array positions 4 and 5. As the sample frequency is about 8 MHz and 32 samples are used for the FFT, the frequency resolution is 8/32 MHz = 250 kHz and the Nyquist frequency is 4 MHz. The following Table 5.1 shows the assignment of real (R) and imaginary (I) values and frequencies to the array positions and bins.

Table 5.1: rFFT Output Assignment

FFT Bin	**Array Position**	**Assignment** (Frequency)	**FFT Bin**	**Array Position**	**Assignment** (Frequency)
1	0, 1	0 Hz, 4 MHz	9	16, 17	R, I @ 2 MHz
2	2, 3	R, I @ 250 kHz	10	18, 19	R, I @ 2.25 MHz
3	4, 5	R, I @ 500 kHz	11	20, 21	R, I @ 2.5 MHz
4	6, 7	R, I @ 750 kHz	12	22, 23	R, I @ 2.75 MHz
5	8, 9	R, I @ 1 MHz	13	24, 25	R, I @ 3 MHz
6	10, 11	R, I @ 1.25 MHz	14	26, 27	R, I @ 3.25 MHz
7	12, 13	R, I @ 1.5 MHz	15	28, 29	R, I @ 3.5 MHz
8	14, 15	R, I @ 1.75 MHz	16	30, 31	R, I @ 3.75 MHz

5.2.1.3.2 Reflection Coefficient Calculation

As the expected digital intermediate alias frequency (after under-sampling down-conversion of the reflected signal) is about 500 kHz (for a sample clock of about 7.99 MHz), array position 4 contains the real value of interest a and position 5 the imaginary value b, this is bin 3. The FFT is also done in the same way for the incident signal, resulting in the real value c and the imaginary value d. These four values are used to calculate the reflection coefficient using equations (52) and (53):

$$\mathrm{Re}\{\Gamma\} = \frac{a \cdot c + b \cdot d}{c^2 + d^2} \tag{52}$$

$$\mathrm{Im}\{\Gamma\} = \frac{b \cdot c - a \cdot d}{c^2 + d^2} \tag{53}$$

The resulting two float values are transmitted to the PC. The reception from the ADC, the rFFT, the calculation of the reflection coefficient and the transmission to the PC are repeated until all the requested measurements were completed or until the stop command was sent by the PC.

The used bin for the reflection coefficient calculation can be changed with a single command, allowing the system to work with other intermediate frequencies. The array position is 2(bin-1) for the real parts and 2(bin-1)+1 for the imaginary parts. Bin 1 usually does not contain useful information for the reflection coefficient calculation as stated before.

5.2.1.3.3 Raw Value Output

The software implements the function to transmit the raw values as read from the GPIOC (and thus the ADC) without the rFFT or the reflection coefficient calculation for debugging of the parallel synchronous transmission. In this case, for every requested measurement, 2x32x16 bit unsigned integer = 128 byte are transmitted to the PC. It becomes clear that for

transmitting the data, the DMA is a reasonable choice, while a maximum of 13 bytes is received (for the rFFT test command), and only in rare occasions.

5.2.1.3.4 rFFT Test

To test the rFFT functioning, a function generates cosine wave forms (using the CMSIS function “arm_cos_f32”) with PC controlled amplitude, frequency and phase and write them in an array with 32 values. Then the function calls the rFFT function and transmits the complete results array (32x32bit float = 128 byte) to the PC.

5.2.1.3.5 Interface to Personal Computer (ST-LINK/V2-1)

The NUCLEO-F446RE is remotely controlled by the PC. No operation is executed without receiving a command. The following commands are possible:

- 'C‘: CONFIG PLL
  The next five bytes encode the frequency controlling registers of the PLL: R (2 bytes), N (2 bytes) and A (1 byte). The PLL configuration is done with these values.
- 'S‘: START DATA
  The next four bytes encode the number of desired measurements as unsigned 32bit integer. The first measurement is started, the results (raw: 2x32x16 bit unsigned integer = 128 byte or reflection coefficient: 2x32 bit float = 8 byte) are sent, then the next measurement and so on until the desired number of measurements was completed. The measurement only starts after the PLL configuration was completed.
- 'E‘: END DATA
  The number of desired measurements is set to zero, no more measurements are done.
- ‘B’: FFT BIN
  The next byte encodes the number of the bin that will be used for reflection coefficient calculation. The array position is bin*2 for the real parts and bin*2+1 for the imaginary parts. This is directly connected to the analyzed frequency. The bin number can be any value between 0 and 15, higher values are limited to 15. This is because the rFFT is limited to a 32 value array in this program (this means, 16 bins).
- 'R‘: RAW DATA
  The next received byte can be zero or one to switch the raw data transmission off or on.
- 'F‘: TEST FFT
  The next twelve bytes encode amplitude, frequency, and phase for the test signal, encoded as 32bit float values each. The device answers with the result array of the rFFT (32x32 bit floats as 128 byte).

The reception of the commands from PC via the USART2 peripheral is interrupt-controlled and uses a switch-case construct in the callback to implement the protocol for low core loading. The USART2 is externally connected with the ST-LINK/V2-1 controller on the NUCLEO-F446RE which works as bridge to the USB and thus to the PC. For the PC, the communication with the application simply works via a virtual COM-port (VCOM).

5.2.2 Personal Computer Software

The personal computer software has the following tasks:

1. Provide an easy-to-use interface for user control of the reflection coefficient measurement system.
2. Allow to receive the data from the reflection coefficient measurement system hardware.
3. Allow to save the received data.
4. Offer possibilities for testing the various communication paths in the system as well as the functioning of the core software components of the microcontroller program (rFFT and reflection coefficient calculation).
5. Provide a simple graphical view of the received data (and to save this view).

For fulfilling the tasks, routines for serial communication, file access and the graphical user interface (GUI) need to be programmed. C# was chosen as language for this PC program, because most necessary functionality (methods) is already available in the .NET library. Thus, the program was based on examples from the library.

5.2.2.1 IDE: Microsoft Visual Studio C#

The language C# is traditionally programmed using the Microsoft Visual Studio IDE [65]. Microsoft offers this IDE for free for university and private users (but a registration is required to use it for more than 30 days).

The programming process is split into a graphical user interface (GUI) design part and an object-oriented code editing part, like in most other visual programming languages. Both are linked via the properties and methods of the GUI objects, especially the event handling methods. The code/GUI files are up- and downwards compatible over the various versions of the Visual Studio from before 2010 on. This was of advantage as the program was begun in a Visual Studio 2010 environment but was further developed in parallel on another machine using a modern Visual Studio (2017). The file exchange between both versions works without any problems in both directions. In the following Figure 5.14, the GUI view of the German version of the Microsoft Visual Studio 2010 IDE is shown.

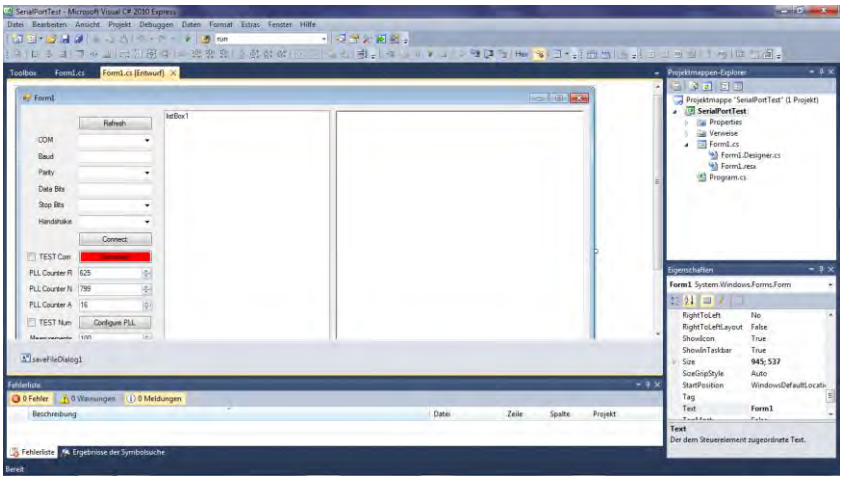

Figure 5.14: Microsoft Visual Studio C# 2010 (GUI Editor)

5.2.2.1.1 NET and MSDN

The Microsoft Developer Network (MSDN) [66] is a community for developers using Microsoft programming languages. There also exists a huge help/documentation section with information regarding the .NET library. There all classes, objects, methods, and properties of the library are described, and can be found either using a hierarchical access or a search function. The documentation also contains a vast amount of code examples and whole programs that demonstrate how to use the classes, objects, methods, and properties in the right way.

5.2.2.2 Virtual COM-Port Communication

The .NET library provides an example for communication via COM-ports (virtual or hardware, for the software this is the same). The example was meant to be a console application, so some adjustments were necessary to make it work in a so-called forms application (this means that the application uses a GUI). The example uses multi-threading to allow for fast reaction on received data, what was kept in the final program. A background process handles the data reception and is divided into different modes, comparable to the reception routine in the microcontroller program. The difference is that in the PC program, the mode is mainly determined by the entries on the GUI and only partly by the received data.

A view of the GUI in Smith chart test mode (see below) is shown in Figure 5.15:

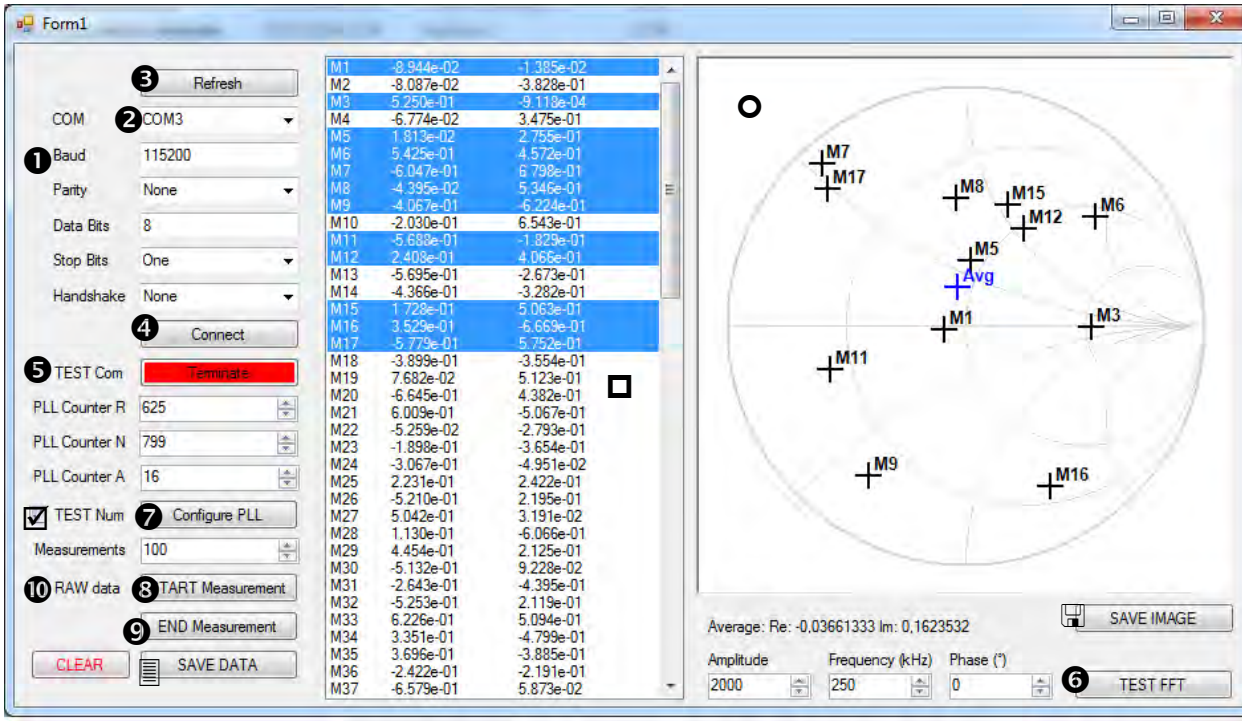

Figure 5.15: PC Software GUI in Mode "TEST Num"

The GUI allows to modify all settings of the COM-port ❶ but is programmed to offer the right values at startup. Only the COM-port to which the measurement system is attached needs to be chosen ❷. If the device is connected after the program startup, the button "Refresh" ❸ needs to be clicked to refresh the COM-port list before the right port can be chosen. Then the button "Connect" ❹ must be pressed to initialize the COM-port and open the communication channel.

A possibility to test the COM communication separately for both directions was implemented. The PC program can be set to the communication test mode by checking the checkbox “TEST Com” ❺. Then the blue button on the microcontroller board can be pressed to send a test message from the microcontroller to the PC. In the other direction (PC to microcontroller) any command can be sent (not using the “Test COM” mode) and the reception be checked using the debug function of the microcontroller IDE. Or the FFT test function ❻ can be used, as this sends back an answer in any case.

Generally, for normal operation, the “TEST Com” ❺ checkbox should not be checked. Then pressing any button sends the according command to the microcontroller. This includes the commands “Configure PLL” ❼, “START Measurement” ❽, “END Measurement” ❾ and “TEST FFT” ❻. The properties for each of these commands are available for editing close to the respective button on the GUI. On changing the state of the checkbox “RAW data” ❿, the raw data on or raw data off commands are sent to the microcontroller and the PC software changes the reception state accordingly. The “FFT Bin” command is not included in the GUI so far but may be included in a later version.

5.2.2.3 Data Display

The received data is asynchronously displayed in a listbox ☐. The format varies according to the mode, but there is always a new line for a new measurement or any single message from the microcontroller.

In normal mode, each line in the listbox begins with an M and the number of the measurement, then the real value and last the imaginary value of the complex reflection coefficient that was measured is displayed. In this mode, the Smith chart display ❍ on the right side will display meaningful graphics when selecting one or more of the measured lines in the listbox with the mouse (and possibly the “Shift” or “Ctrl” button of the keyboard for multiple selection). The lines do not need to be consecutive; any number of distributed values is possible. Selected entries are converted to numbers and displayed in the Smith-chart, marked with the measurement number for reference. The graphics functionality was implemented based on earlier experience with image generation and research about the listbox options.

In raw value transmission mode, the 2x32 unsigned 16bit integer numbers that are received from the microcontroller are displayed in hexadecimal format, separated by commas, also in a single line for each measurement. In this mode, the Smith chart display on the right will not display meaningful content.

Also, in the FFT test mode, the Smith chart will not display anything meaningful. The listbox will display “FFT” and the number of the transfer as well as the FFT test data like it was sent to the microcontroller (3 values separated by commas). Then more 32 values separated by commas, encoding the FFT output like described in section 5.2.1.3.1 are displayed.

In communication test mode, the text message sent by the microcontroller saying that the button was pressed is displayed in the list box, after a “T” and the test run number.

The listbox allows scrolling in any direction where there is data.

Finally, for testing the Smith chart functionality, there is another checkbox “TEST Num” ☑. It generates random complex numbers inside the possible range and in the right format when the “START Measurement” button is pressed and fills them into the listbox. They can be selected and displayed in the Smith chart just like real measurement values.

5.2.2.4 Saving Data (and View)

The standard file save dialog was included in the program, for saving the contents of the listbox in a text-file (comma separated values “.CSV”) as well as for saving the Smith-chart view in an image file (Compuserve bitmap “.GIF”). The file save dialog opens after pressing either the „SAVE DATA“ 🗎 or the „SAVE IMAGE“ 🖫 button, with options adjusted to either case. For implementing the functionality, another example from the .NET library was used together with some experience from former projects. In Figure 5.16 a view of the file save dialog is shown.

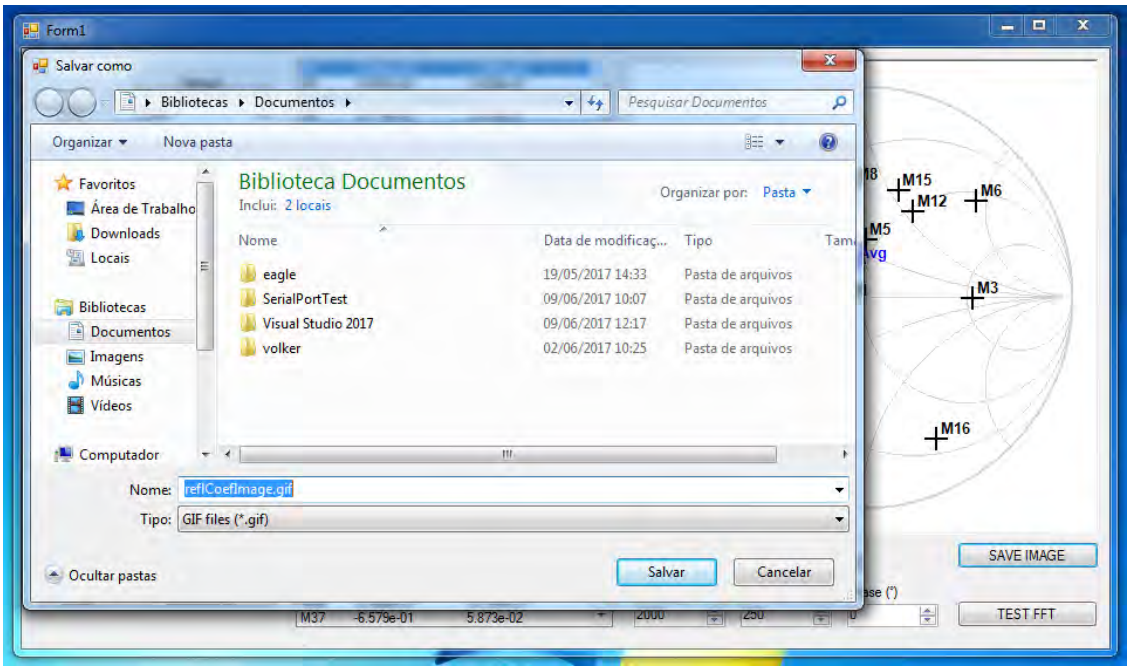

Figure 5.16: Save File Dialog of PC Software

5.2.2.5 Various Testing Possibilities

Not only for prototyping it is important to implement means of testing various parts of the hardware and software, as a reliable system can only be made when errors in the implementation can be found and fixed. In this case, the PC program allows to test/debug the following modules of the project:

- Communication between Device and PC: Test message transmission, display via PC-program or debugger.
- Programming of the PLL: a logic analyzer demonstrated the compliance of the SPI message sent by the Microcontroller to the PLL chip ADF4360-9 with the datasheet of the PLL after the PC program gives the according command.
- Parallel data reception from ADC: the raw data can be transmitted to the PC program, displayed and saved. This was useful for debugging the transmission path.

- Functioning of the rFFT and reflection coefficient calculation. For the latter, both input data arrays of the rFFT can be modified using the debugger of the microcontroller IDE and the results can be recorded and saved by the PC software. This way, input waves with various amplitude and phase relationships were generated and the results were verified.

Generally, all implemented functionality of microcontroller and PC program was tested when possible and bugs were fixed whenever they were found.

It remains to be said that the PC program still has a problem with terminating the background thread. This only works right when the red “Terminate” button is pressed before closing the program. After pressing the “Terminate” button, no more communication is possible. Efforts to include this functionality into the method that is called when the program is closing were not successful. If the “Terminate” button is not pressed before closing the program once the connection is active, the program will hang up on closing.

6 Characterization

After the basic assembly of the system (see assembly documentation in Appendix A.2) and the clarification of the availability of a 400 MHz RF source, the commissioning of the electronics was started. The objective for the characterization was not to ultimately test every single component of the proposed system, but to get the system running as a whole and to gain information about its general performance as a reflectometer and possible drawbacks.

To get the system up and running, first the main components (PLL and ADC) were tested. Because the ADC cannot work without a sample clock, the first component to be tested was the PLL frequency synthesizer. After this, the ADC could be tested and finally it was possible to test the reflectometer with known impedances. Together with the main components, supporting components were tested as appropriate.

6.1 Test of PLL and output Frequency

6.1.1 Test Setup

The source used for these function tests is the tracking generator of the HP 8594E Spectrum Analyzer ❶ available at Hahn-Schickard Stuttgart. The output of the tracking generator was directly connected to the forward input of the system PCB ❷. System PCB and Nucleo-board ❸ were connected via the ribbon cable ❹ and the adapter PCB ❺, and the Nucleo-board was connected to the PC ❻ (a notebook) where the control software was running via USB ❼. The system PCB was powered using a Hameg HM7042-5 laboratory power supply ❽. The Test Setup is shown in Figure 6.1 and Figure 6.2.

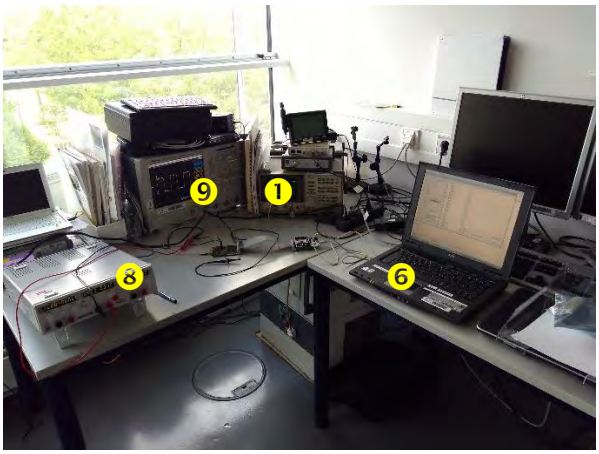

Figure 6.1: Test Setup - Overview

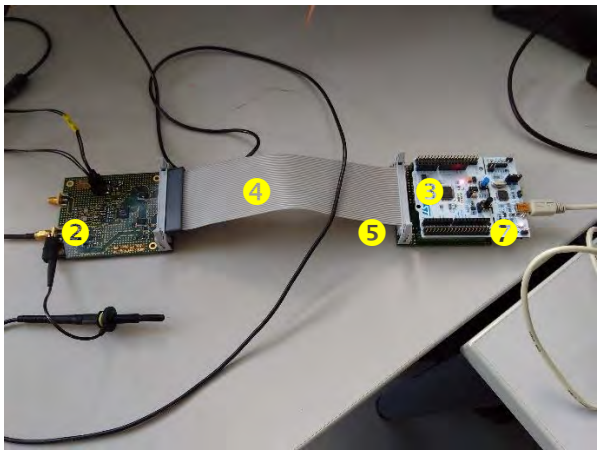

Figure 6.2: Test Setup - Closeup

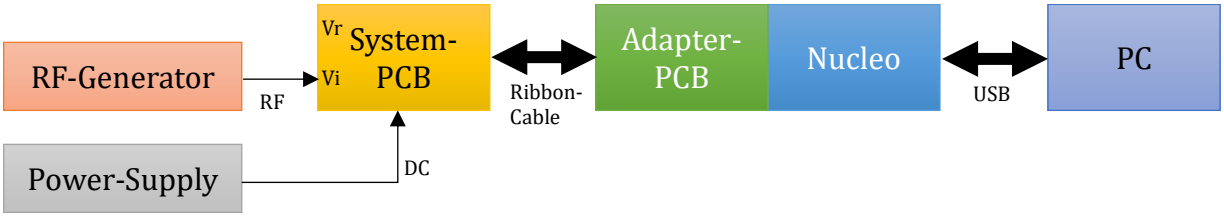

Figure 6.3: Block Diagram of Test Setup 1

6.1.2 Test Execution

The control software was used to program the PLL chip registers. Finally, a Tectronix TDS5104B oscilloscope ❾ was used to measure the output signal of the PLL chip after the buffer (i.e., the signal that should arrive at the sample clock input of the ADC).

6.1.3 Test Results

The output frequency of the PLL chip after programming was found to be 7.99 MHz (measurement point 1☝ shown in Figure 6.4). The signal shape was a 50% duty cycle square wave as expected.

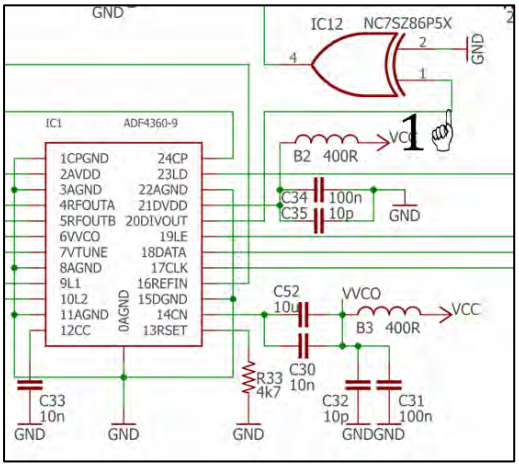

Figure 6.4: Measurement Point for PLL Output

So, the Pre-Divider, the PLL, the SPI, the PLL driver in the microcontroller and the respective part of the control software work as expected. The phase noise of the PLL was not measured here, as the feeding and measurement equipment is quite aged, and the phase noise contribution is not known.

6.2 Test of Forward Channel of ADC and Measurement Chain

The test setup was the same as in section 6.1.1.

6.2.1 Test Execution

Test of the forward channels of the ADC by direct 400 MHz input of the maximum possible amplitude (about 2.7dBm) in under-sampling mode and recording of the sampled raw values with the PC, without using a directional coupler. This tests the whole board, microcontroller software and PC software basic function. Not tested is the ADC input channel for the reflected signal. This was later tested in the complete setup measuring first reflection coefficients.

6.2.2 Test Results

In a first test run, some hardware problems were found.

- The data clock for clocking in the parallel data from the ADC to the microcontroller was not transmitted, therefore it was not possible to read-in the data. Further investigations showed that there was a pin not soldered at the clock gating circuitry

comprising of a non-inverting D-flipflop and an AND-gate. The pin in question was pin 2 of the AND-gate IC13, used as enable input, so basically the clock was never enabled. This problem was easily solved using a standard solder iron with a fine tip.

- The ADC (IC2) seemed to be working basically (signals were obtained at the system PCB outputs), but one of the 15 data lines of the ADC did not show activity. This was also due to a pin that had been soldered incorrectly during the reflow process and was solved the same way as the other loose pin. It might be possible that some other data lines have unreliable contacts, what will be considered if the results of the tests with the directional coupler show strange results.

After the hardware problems were fixed, it was possible to transmit data to the PC via the Nucleo board. As there was no input data on the reflected signal input of the ADC, the obtained reflection coefficients did not make sense, as expected.

6.3 Measurements with known Impedances

The characterization was concluded by measuring known impedances and evaluating measurement performance and errors.

6.3.1 Measurement Setup

The source used for the measurements is the tracking generator of the HP 8594E Spectrum Analyzer. The output of the tracking generator was directly connected to input of the proprietary PA. The PA is an unprotected proprietary device designed and characterized in [16]. The output of the PA was connected to the input of ZFBDC20-62HP-S bi-directional coupler. Its coupled outputs were connected to the respective inputs of the system PCB. System PCB and Nucleo-board were connected via the ribbon cable and the adapter PCB, and the Nucleo-board was connected via USB to the PC (a notebook) where the control software was running. The system PCB and the PA were powered using a Hameg HM7042-5 laboratory power supply. A block diagram of the measurement setup is shown in Figure 6.5 and a photograph in Figure 6.6.

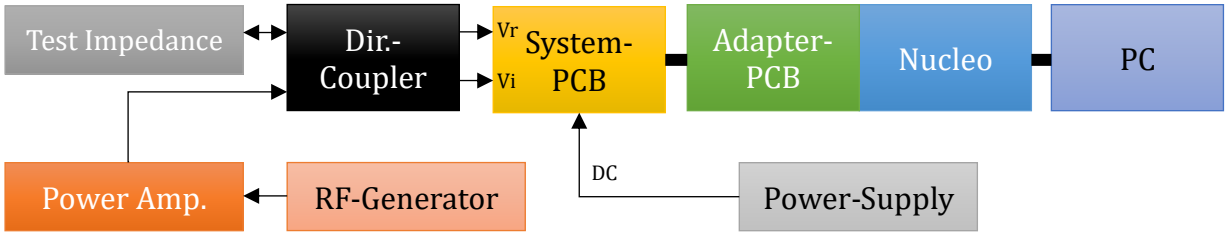

Figure 6.5: Block Diagram Test Setup 2

Different known cable lengths and termination impedances were attached to the mainline output of the bi-directional coupler and subsequently used to characterize the measurement block. These reference impedances were also measured using a Vector Network Analyzer (VNA). This was done because there was no impedance tuner available at the site of measurement, and no resources for acquiring one.

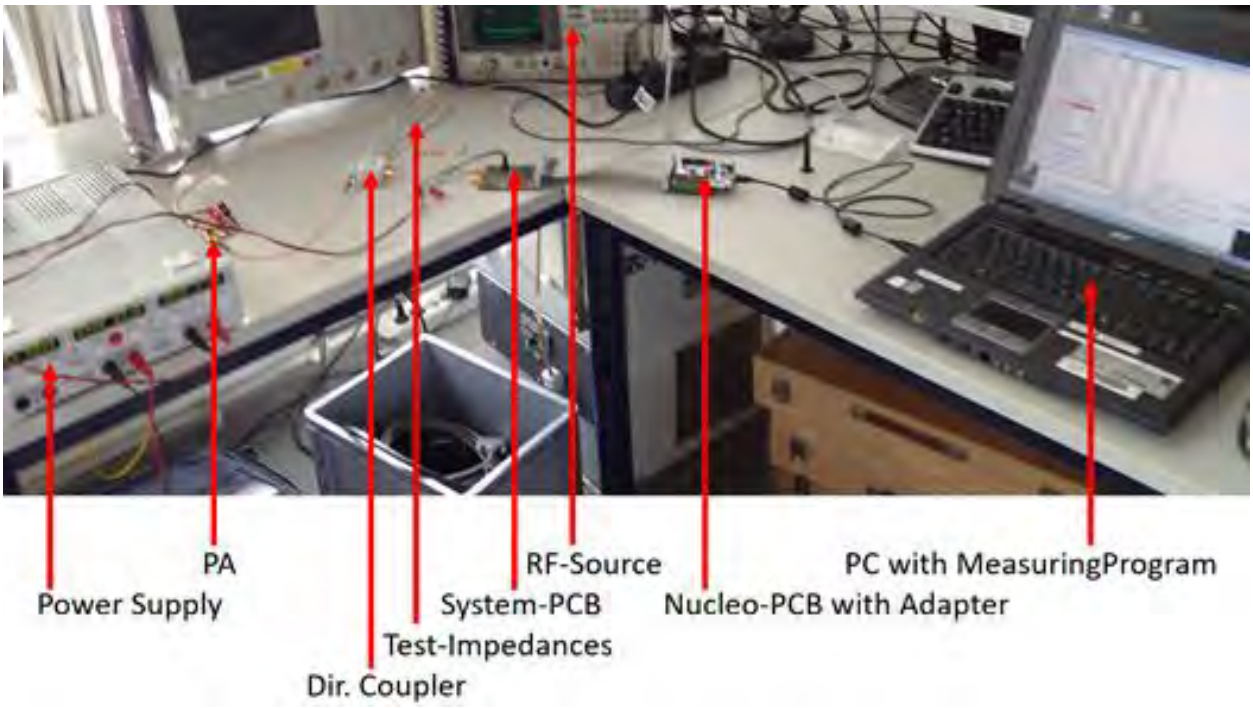

Figure 6.6: Measurement Setup for Reflection Coefficient Measurements

6.3.2 Measurement Results

The names of the conditions as stated in the first column of Table 6.1 comprise of the cable lengths in millimeters (172, 207 and 241 mm) and the attached termination impedance: 0 Ω, 16.6 Ω, 50 Ω, 150 Ω, a connector without attached impedance (conOpen), or a completely open socket (open). The difference between "conOpen" and "open" is the length of the connector and connector parasitic inductance and capacitance, what presents a slightly different impedance to the reflectometer than the open socket.

The next two columns are the reflection coefficients for these conditions as measured by a Vector Network Analyzer (VNA), then the average (over 100 measurements) values measured by the Device under Test (DUT), then the reflection coefficients derived from the DUT values by using the short-open-load (SOL) correction algorithm (also averages of 100 measurements). The according SOL MATLAB script based on [67] is attached in Appendix A.3. Then follows the evaluation of the error E = (SOL-VNA) and the standard deviation over 100 measured values of the reflection coefficient for each condition. The value used for evaluation is the magnitude "mag" of the complex numbers (error and standard deviation).

Table 6.1: Complete Measurement Results including SOL-Correction

Condition	S11 VNA		S11 DUT		S11 SOL-Corrected		Average Error S11 SOL-VNA			Standard Deviation S11 SOL			Comment
length-imped.	real	imag	Real	imag	real	imag	real	imag	mag	real	imag	mag	
0-Open	1,0000	0,0000	-0,8996	0,4695	1,0000	0,0000	0,0000	0,0000	0,00%	0,00052	0,00060	0,001	SOL ref
172-0 Ω	0,3000	-0,9300	0,2155	-0,2078	-0,1871	-0,0521	-0,4871	0,8779	100,40%	1,48477	0,91680	1,745	sync lost
172-16.6 Ω	0,1100	-0,4700	0,2125	0,3385	0,1201	-0,4698	0,0101	0,0002	1,01%	0,00070	0,00071	0,001	
172-50 Ω	-0,0400	0,0050	0,0598	-0,1647	-0,0401	-0,0008	-0,0001	-0,0058	0,58%	0,00056	0,00058	0,001	
172-150 Ω	-0,2100	0,4700	-0,0399	-0,6681	-0,2273	0,4624	-0,0173	-0,0076	1,89%	0,00091	0,00122	0,002	
172-conOpen	-0,4200	0,9050	-0,1651	0,3051	0,3821	-0,2406	0,8021	-1,1456	139,85%	2,35908	2,33429	3,319	sync lost
172-open	-0,5800	0,8150	-0,0048	0,6004	0,4223	-0,5411	1,0023	-1,3561	168,63%	1,55165	2,78676	3,190	sync lost
207-0 Ω	-0,4600	-0,8600	0,9131	-0,2986	-0,8006	-0,3931	-0,3406	0,4669	57,80%	0,10782	0,09788	0,146	sync lost
207-16.6 Ω	-0,2700	-0,4100	0,4924	-0,0499	-0,3151	-0,3347	-0,0451	0,0753	8,77%	0,01015	0,01003	0,014	failure starts
207-50 Ω	-0,0300	0,0250	0,0380	-0,1775	-0,0299	0,0211	0,0001	-0,0039	0,39%	0,00055	0,00060	0,001	
207-150 Ω	0,1800	0,4700	-0,4032	-0,4564	0,1801	0,4719	0,0001	0,0019	0,19%	0,00064	0,00073	0,001	
207-conOpen	0,3600	0,9400	-0,8181	-0,7927	0,3646	0,9429	0,0046	0,0029	0,55%	0,00083	0,00090	0,001	
207-open	0,1800	0,9950	-0,6731	-0,9316	0,1800	0,9950	0,0000	0,0000	0,00%	0,00112	0,00103	0,002	SOL ref
241-0 Ω	-0,9350	-0,2800	0,9563	0,4305	-0,3727	-0,9808	0,5623	-0,7008	89,85%	0,08669	0,09054	0,125	sync lost
241-16.6 Ω	-0,4900	-0,1000	0,4987	0,3341	-0,0945	-0,6306	0,3955	-0,5306	66,18%	0,05086	0,05449	0,075	sync lost
241-50 Ω	-0,0100	0,0300	0,0156	-0,1735	-0,0100	0,0300	0,0000	0,0000	0,00%	0,00055	0,00053	0,001	SOL ref
241-150 Ω	0,4600	0,1800	-0,4930	-0,0306	0,4599	0,1793	-0,0001	-0,0007	0,07%	0,00046	0,00047	0,001	
241-conOpen	0,9250	0,3600	-1,0451	0,0726	0,9280	0,3577	0,0030	-0,0023	0,38%	0,00044	0,00042	0,001	
241-open	0,8400	0,5350	-1,0629	-0,1437	0,8465	0,5301	0,0065	-0,0049	0,82%	0,00050	0,00043	0,001	

For reference, the reflection coefficients will also be shown as chart in Figure 6.7:

Figure 6.7: Chart of Measured Reflection Coefficients

6.3.3 Analysis and Discussion

It becomes clear that the errors of the averages over 100 measurements are typically considerably below 2% when the standard deviation of the measurement is in the order of 0.002 or lower. It was also observed that in cases with higher standard deviation, that produced invalid measurements, there were strong harmonics present at the input of the pre-divider.

6.3.3.1 Most probable Explanation

The harmonics are produced by the unprotected PA under extreme load impedance conditions due to nonlinear effects, as there is no other device in the signal path that could produce these harmonics. The consequence of the harmonics are false trigger events of the pre-divider, resulting in incorrect sampling frequencies and thus no well-defined relationship between signal frequency and sample clock. This leads to high sampling noise and thus to a high standard deviation. Accordingly, the under-sampling down-conversion did not work or begin to fail in the cases with a high standard deviation (0.014 or higher). Under these circumstances, no correct measurement is possible, so high error magnitudes are not surprising.

6.3.3.2 Relevance of the Problem and Countermeasures

In the final automatic impedance matching system, these extreme conditions should normally not occur, as they are corrected before they become severe. Additionally, the output power of the PA could be reduced when such a condition is recognized by the software (effortlessly identified by means of the standard deviation), reducing harmonics and allowing ongoing measurements until the condition ends. However, as this reduces the transmitted power, it can have negative effects on the main function of the transmission system, the communication.

For the open loop measurements however, the problem could be resolved using an RF isolator between PA and directional coupler, or by replacing the directional coupler with an RF circulator. Both measures would prevent the reflected signal from reaching the PA, therefore also avoiding the generation of harmonics. However, RF circulators and isolators are often made from ferrites, and are generally difficult to miniaturize [16]. Active circulators exist and can be miniaturized but might suffer nonlinear effects due to reflected power themselves.

Finally, for both applications, the input signal for the PLL could be derived after the narrow-band AA-filter of the ADC, or a third narrow band filter can be added in front of the PLL pre-divider, consequently reducing harmonics at the input of the PLL. As the system uses narrow band filters anyway, this might be the best solution with least disadvantages and could even allow better performance when modulation is present, given the signal passes the filter.

6.3.3.3 Considerations for Modulation

Modulated signals should not have such a destabilizing impact on the system, because the pre-divider includes a comparator input which effectively removes amplitude modulation (AM). The current system uses the Schmitt-Trigger input of the D-flipflop for simplicity, effectively limiting the possible modulation depth of the AM, but a system optimized for AM inputs can be derived by including a separate comparator or an amplifier with a limiter. High AM speed and modulation depth however may lead to additional noise in the measurements, but not to a failure of the under-sampling down-conversion. For sufficiently distributed modulation input, averaging should still produce reasonable output in this case. Similar considerations hold true for frequency modulated (FM) inputs, because the PLL can follow the modulation if the bandwidth is not too high, so that the sample clock is tracking the signal. This is a design property of the PLL and can be optimized separately. The properties of the used PLL configuration are shown in Appendix A.1.

Digital modulation techniques however, such as quadrature amplitude modulation (QAM) or phase modulation, might be problematic due to their wide bandwidth.

Unfortunately, a more advanced signal generator that would have allowed tests with modulated input to the system originally available at the test site (Hahn-Schickard Stuttgart) was found inoperative and repair was not possible anymore due to the age of the device. The replacement used for the tests above was the tracking generator of a spectrum analyzer, that does not offer modulated output capabilities.

6.3.3.4 Performance Comparison with Published PCB-Setups

As shown in Table 6.2, the proposed system performed very well compared with other published works on PCB setups.

Table 6.2: Performance Comparison

This Work:	**[16]**	**[68]**	**[69]**
< 2%*	28%	25%**	37%

* When Under-Sampling Down-Conversion is working correctly.
**derived from given Smith chart

7 Optimized Algorithm for UHF Software Defined Reflectometer

In the current work of the author, different aspects of a software defined reflectometer using under-sampling down-conversion in the inevitable analog-digital converter were highlighted, such as the under-sampling theory and sample frequency generation, as well as the design of the minimum necessary hardware for such a system and measurement and simulation results. An article about these findings was also presented at the EUMW 2019 in Paris [70] (also see Appendix A.4).

However, so far only one algorithm for the software part of the system was investigated, using the well-known fast-Fourier transform (FFT) in its ARM CMSIS implementation "arm_rfft_f32". Furthermore, precision was optimized by synchronizing the signal frequency with the sample clock, eviting problems with unsynchronized signals.

In the context of this work, further investigations regarding an optimized algorithm for the digital signal processing of raw sinusoidal signals were and are still carried out under the supervision of the author.

7.1 Implementation and Selection of Algorithms for Optical Ranging Sensors

The results of the recent bachelor work “Implementation and Selection of Algorithms for Optical Ranging Sensors“ [71] can easily be transferred to general software defined reflectometers, as the basic setup assumed for the optical ranging sensors in question uses continuous wave signals, one forward and one reflected signal. So basically, the setup resembles an RF reflectometer with an additional optical path. Also, the computing platform for the analyzed digital signal processing algorithms was chosen for compatibility with both the platform of [70] and the platforms normally used for such purposes at Hahn-Schickard Stuttgart, the ARM Cortex-M4 STM32 family of microcontrollers, here an STM32F446 microcontroller.

In [71], a comprehensive literature research for algorithms that extract either the complex amplitudes or magnitudes and phases from error-prone sinusoidal signals relative to a reference signal at a given frequency was carried out. Subsequently, the four most relevant algorithms were implemented and compared by means of non-ideal test signals generated in MATLAB and an evaluation matrix.

The following comparison criteria were evaluated:

- Computational efficiency by means of memory usage and execution time,
- Immunity against noise in phase and amplitude, and
- Immunity against (slight) deviation in signal frequency from the assumed value.

As a comparison reference, a fifth algorithm like the setup used so far in this thesis was implemented, the ARM CMSIS rFFT-function mentioned above. For better comparability with the other algorithms in respect of frequency deviation, the synchronization scheme used so far was replaced by windowing [72], namely a Hann window. The Hann window is basically cosine-shaped, converging to zero at the edges, and should reduce the effects of missing synchronization, such as poor accuracy in amplitude and phase measurement due to the implicit rectangular window when not explicitly using a window function.

7.1.1 Implemented Algorithms

As stated before, a total of five algorithms were implemented on the STM32-microcontroller and tested using nonideal signals. The algorithms are covered in [69], so there will only be given the references to the originating papers here:

- RGN: Recursive Gauss-Newton [73]
- Mikhotin's Theorem [74]
- Three-parameter sine fitting algorithm [75]
- All-phase spectrum analysis and convoluted Hann windows [76]
- rFFT with Hann window (the comparison reference)

7.1.2 Test Scheme

The signals were software generated in MATLAB on a Personal Computer (PC) based on a general sinusoidal signal equation [71] and a set of controlling variables such as amplitude noise, phase noise and frequency, ensuring a maximum comparability of the input for the algorithms.

The numeric samples of the software generated signals were transferred to the microcontroller via USB and UART and the results were transferred back the same way to the PC, where they were stored and evaluated.

Furthermore, the basic accuracy of the algorithms was evaluated with (near) ideal test signals, and the execution time and memory usage of each algorithm was measured.

More tests will be carried out in the future with additional signal nonidealities such as harmonic content, or modulation artifacts and the like, but as these can be seen as a kind of noise, the impact on the results is expected to be low, especially for the FFT group of algorithms.

7.1.3 Results

The measured values were compiled into an evaluation matrix [71]. The chosen weights of the criteria were as shown in Table 7.1:

Table 7.1: Criteria and weights for evaluation of algorithms

Criteria	Weights
Robustness against amplitude noise	4
Robustness against phase noise	4
Robustness against frequency deviation	4
Basic accuracy	4
Execution time	2
Memory usage	2

Based on these weights, [71] came up with the following ranking of the implemented algorithms:

1. Recursive Gauss-Newton
2. Mikhotin's theorem *AND* Three-parameter sine fitting
3. rFFT with Hann window
4. All-phase spectrum analysis and convoluted Hann windows

Further research will be conducted in this field, with thoroughly automated tests and thus a considerably higher amount of statistical data, tested algorithms, and criteria. However, it already became clear that the classical FFT and its derivatives might not be the final answer for the software part of the software defined reflectometer.

This application additionally necessitates an evaluation for the compatibility of the sinusoidal analysis algorithm with the calculation of the reflection coefficient itself. Some algorithms such as Mikhotin's theorem calculate a phase difference and the two magnitudes of the two input signals, what involves more complex math for the further calculations, others output two complex amplitudes, enabling the use of the same subsequent math as originally.

8 Foundation for Closing the Loop

As stated before, one promising application of the software defined reflectometer is as measurement block in an automatic impedance matching system (AIMS). Such a system comprises of two more blocks, namely the control block and the variable impedance matching network.

For reference, Figure 1.2 from the introduction will be repeated here (Figure 8.1):

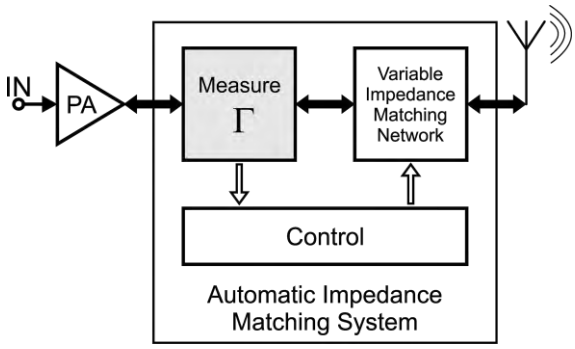

Figure 8.1: Automatic Impedance Matching System

8.1 Variable Impedance Matching Network

As described before, the variable impedance matching network (IMN) already is well covered in literature. A very promising, lumped approach is the Pi-Network comprising of two grounded adjustable capacitors and one not grounded adjustable inductor. For a known frequency, lumped impedance inverters (also a CLC-Pi-Network) can be inserted on both sides of the inductor. This way, also the inductor can be replaced by a grounded adjustable capacitor. Instead of one of the three grounded adjustable capacitors, a bank of few values of capacitors with RF-switches to ground can be used. This was already described on p.66/67 in [16].

For this network, matching circles [4] were obtained for the base state, this is the switch configuration that would be correct for no (or only negligible) mismatch [16]. Based on these matching circles, a first control algorithm was implemented.

8.2 Complex Control Algorithm Theory

For closing the loop in an automatic impedance matching system using the complex reflection coefficient information obtained from the described reflectometer, a complex control algorithm (CCA) must be designed to actuate the variable IMN.

8.2.1 Previous Algorithm

The CCA described in [16] can be considered promising, because it already operates in two dimensions (real and imaginary) to use the reflection coefficient information effectively. Instead of only using the sign of the real and imaginary part and a binary search scheme as in [77], the full available information is utilized in a straight-forward manner

without requiring a complicated search. Furthermore, [16] implements matching regions (circles) of allowed mismatch together with a geometric hysteresis (avoiding fluttering between the states) rather than ideal points, both desirable properties for an algorithm working with a limited number of matching states and with switches, either electronic or as micro electro-mechanical systems (MEMS).

However, the original implementation required to return to the variable IMN base state switch configuration for measuring the reflection coefficient and then actuating the switches accordingly to reach the matched state. This involves a short moment of possibly considerable mismatch seen by the further blocks of the system such as the power amplifier (PA). Therefore, the PA still needed to be hardened to mismatch conditions.

8.2.2 Proposed Solution

The CCA described in [16] can easily be improved by calculating the circles not only for the base state, but also for every other possible matching state. This is done by calculating the reflection coefficient matching circles for the same switch configurations (output impedances), but different load impedances. This results in the square number of the original matching circles, as for every state there must be calculated circles for itself and every other state.

An effective implementation is therefore a square matrix of circle definitions (centers C_{mn} with x/real and y/imag coordinates and radiuses R_{mn}, an example is shown in Table 8.1). One dimension of this square matrix is then the current matching state (CMS), the other dimension is the test matching state (TMS).

The algorithm then stores the active CMS position (index number) and runs through the line vector of TMS, comparing each of the TMS circles to the current reflection coefficient to determine the next CMS. Is the current reflection coefficient inside one of the checked TMS matching circles (distance smaller than radius) and outside of the CMS currently matched circle (grey highlight in the example) that can be found on the main diagonal of the square matrix (denoted in bold italic in the example), then the new CMS index is equal to the found TMS index. Otherwise, the CMS index stays the same. A flow-chart of the algorithm is shown in Figure 8.2.

Table 8.1: Square matrix of matching circles

Example for 5 possible impedance matching network configurations / matching states		**TMS** (test matching state)				
		1	2	3	4	5
CMS (current matching state)	1	***C_{11X}, C_{11Y}, R_{11}***	C_{12X}, C_{12Y}, R_{12}	C_{13X}, C_{13Y}, R_{13}	C_{14X}, C_{14Y}, R_{14}	C_{15X}, C_{15Y}, R_{15}
	2 *(active)*	C_{21X}, C_{21Y}, R_{21}	***C_{22X}, C_{22Y}, R_{22}***	C_{23X}, C_{23Y}, R_{23}	C_{24X}, C_{24Y}, R_{24}	C_{25X}, C_{25Y}, R_{25}
	3	C_{31X}, C_{31Y}, R_{31}	C_{32X}, C_{32Y}, R_{32}	***C_{33X}, C_{33Y}, R_{33}***	C_{34X}, C_{34Y}, R_{34}	C_{35X}, C_{35Y}, R_{35}
	4	C_{41X}, C_{41Y}, R_{41}	C_{42X}, C_{42Y}, R_{42}	C_{43X}, C_{43Y}, R_{43}	***C_{44X}, C_{44Y}, R_{44}***	C_{45X}, C_{45Y}, R_{45}
	5	C_{51X}, C_{51Y}, R_{51}	C_{52X}, C_{52Y}, R_{52}	C_{53X}, C_{53Y}, R_{53}	C_{54X}, C_{54Y}, R_{54}	***C_{55X}, C_{55Y}, R_{55}***

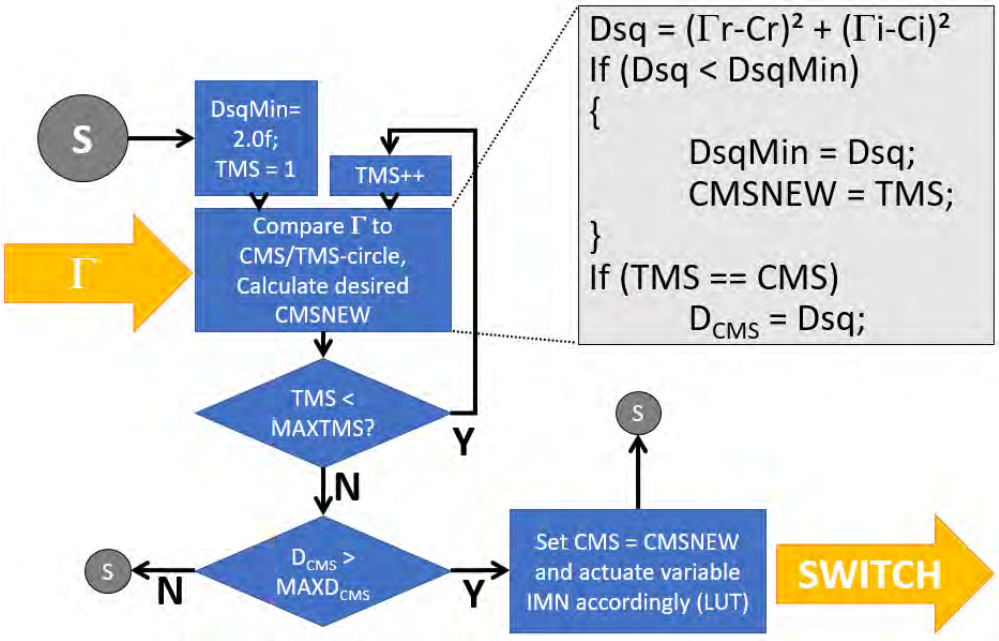

Figure 8.2: Flow-Chart of Proposed Complex Control Algorithm

In other words, the problem of controlling the IMN switches according to the measured reflection coefficient is reduced to geometric distance calculation, limit checking, and a lookup table like in [16]. However, with this new design, the AIMS is no longer required to return to the base state of the IMN for measurement, it can actuate the switches correctly for any new configuration from any given state of the IMN. This promises single cycle tuning instead of repeated refining of the tuning process as in [78].

A future study could implement the IMN and the new CCA and collect measurement data on the complete system. Also, the matching circles' centers and radiuses that are stored in the square matrix lookup table can be fine adjusted to measured values instead of the calculated ones. The approach, as it is based on general matching circle theory, can easily be applied to higher frequencies, as then only the circles must be recalculated (or measured).

9 Conclusion

A new software-defined approach for a reflectometer based on under-sampling down-conversion was developed. Design considerations, simulation- and measurement results were also presented.

Simulations and measurements showed a good accuracy for such a simplified system, mostly better than 2%, when the under-sampling down-conversion was operative, recognizable by a standard deviation of the measurement results of 0.002 or below.

However, a failure of the under-sampling scheme was observed under certain severe load mismatch conditions. The failure of the under-sampling was caused by harmonic frequencies that were present at the input of the sample clock generation block. With high probability, these harmonics come from nonlinear effects in the unprotected power amplifier. These failure cases can easily be identified by higher standard deviation of the results (0.014 or higher were measured).

Possible countermeasures against the effects of the harmonics were discussed, such as protecting the power amplifier, filtering out harmonics before the sample clock PLL and others. In an automatic impedance matching system though, these conditions are unlikely to occur due to their extreme nature and the continuous adjusting operation of the system. However, these and other hardware tasks such as designing a miniaturized directional coupler for the reflectometer can be part of a future research project.

The proposed reflectometer will perform well in situations where there is only one frequency present in the measurement signal or where there is a clear separation between the carrier and the side bands (then the carrier can be used, and the side bands are removed by the anti-alias band-pass filters). So, it can already be applied in measurement systems for material characterization. The use of a software-defined approach is also advantageous when there is already a digital signal processor and a fast ADC present such as software-defined radio. Then the additional cost for applying a software-defined reflectometer is insignificant, also regarding energy consumption.

The proposed software-defined reflectometer will be less beneficial in systems that contain few digital circuits and rely on analog signal processing, as the application is more complex. Also, the behavior of the reflectometer when subjected to modulated inputs should still be investigated for the application in an automatic impedance matching system.

The foundation for closing the loop in an automatic impedance matching system using this reflectometer as measurement block was laid, with a hands-on description of the steps to be done for achieving this goal. A future work can include implementing the proposed complex control algorithm together with an impedance matching network and concluding closed loop measurements.

Finally, future studies could deepen the insight on other possible parameter estimation algorithms for the software part of the system, as was already begun under supervision of the author.

References

[1] A. Varghese and D. Tandur, “Wireless requirements and challenges in Industry 4.0,” 2014 International Conference on Contemporary Computing and Informatics, Electronic ISBN: 978-1-4799-6629-5, November 2014.

[2] J. Toftgard and S. Hornsteth, "Effects on portable antennas of the presence of a person," IEEE Transaction on Antennas and Propagation, Vol. 41, No. 6, pp. 739-746, June 1993.

[3] R. A. Sadeghzadeh, M. S. Abrishamian and N. J. McEwan, "Effect of user head on mobile telephone handset antenna impedance and head internal fields distribution," Procedures 24th European Microwave Conference, pp. 603-606, September 1994.

[4] K. Brito and R. Lima, “Impedance network for an automatic impedance matching system,” 2007 Asia-Pacific Microwave Conference, pp. 1-4, December 2007.

[5] J. Madic, P. Bretchko, Shuyun Zhang, R. Shumovich, R. McMorrow, "Accurate power control technique for handset PA modules with integrated directional couplers," 2003 IEEE MTT-S International Microwave Symposium Digest, Print ISBN: 0-7803-7695-1, June 2003.

[6] C. Sánchez-Pérez, D. Sardin, M. Roberg, J. de Mingo and Z. Popović, “Tunable outphasing for power amplifier efficiency improvement under load mismatch,” Microwave Symposium Digest (MTT), IEEE MTT-S International, pp. 1-3, June 2012.

[7] Han Lim Lee, Won-Gyu Lim, Kyoung-Sub Oh, and Jong-Won Yu, “24 GHz Balanced Doppler Radar Front-End With Tx Leakage Canceller for Antenna Impedance Variation and Mutual Coupling,” IEEE Transactions on Antennas and Propagation, Vol. 59, No. 12, pp. 4497-4504, December 2011.

[8] I. Aoki, ”Fully integrated CMOS power amplifier design using the distributed active-transformer architecture,” IEEE J. Solid-State Circuits, Vol. 37, No. 3, pp. 371-383, March 2002.

[9] Robson N. De Lima, B. Huyart, E. Bergeault and L. Jallet, “MMIC impedance matching system,” IEEE Electronics Letters, Vol. 36, No 16, pp. 1393-1394, August 2000.

[10] Raisa G. Pesel, Sara S. Attar, Raafat R. Mansour, "MEMS-based switched-capacitor banks for impedance matching networks," 2015 European Microwave Conference (EuMC), Electronic ISBN: 978-2-8748-7039-2, September 2015.

[11] Maha Added, Noureddine Boulejfen, "Variable impedance matching network based on varactor diodes," 2015 IEEE 15th Mediterranean Microwave Symposium (MMS), Electronic ISBN: 978-1-4673-7602-0, November 2015.

[12] B. Xiong, K. Hofmann, "Binary search algorithm for adaptive impedance matching network," Electronics Letters Vol.: 52, Issue: 9, PP.: 714-716, April 2016

[13] D. Lauder and Y. Sun, "An Overview of Automatic Antenna Impedance Matching for Mobile Communications," 2020 European Conference on Circuit Theory and Design (ECCTD), Sofia, Bulgaria, pp. 1-4, doi: 10.1109/ECCTD49232.2020.9218350, September 2020.

[14] S. Julrat and S. Trabelsi, "Portable six-port reflectometer for determining moisture content of biomass material," IEEE Sensors Journal, Vol. 17, No. 15, pp. 4814-4819, August 2017.

[15] M. S. McKeown, S. Jurat, S. Trabelsi and E. W. Tollner, "Open transverse-slot substrate-integrated waveguide sensor for biomass permittivity determinantion," IEEE Transactions on Instrumentation and Measurement, Vol. 66, No. 8, pp. 2181-2187, August 2017.

[16] Volker Kible, “An UHF reflection coefficient measurement block based on injection locking and quadrature amplitude demodulation,” CDD 621.381 32, Universidade Federal da Bahia, Escola Politécnica, Salvador, December 2016.

[17] F. Meng, A. van Bezooijen and R. Mahmoudi, “A mismatch detector for adaptive antenna impedance matching,” 36th European Microwave Conference, pp. 1457-1460, September 2006.

[18] S. Kousai, K. Onizuka, T. Yamaguchi, Y. Kuriyama and M. Nagaoka , “Polar antenna impedance detection and tuning for efficiency improvement in a 3G/4G CMOS power amplifier,” ISSCC Dig. Tech Papers, pp. 58-60, February 2014.

[19] J. Moritz, D. Lauder and Y. Sun, “Measurement of HF antenna impedances,” IEE Colloquium on Frequency Selection and Management Techniques for HF Communications, pp. 14/1-14/7, March 1999.

[20] M. Gillick, I. D. Robertson, J. S. Joshi, "A 12-36 GHz MMIC 3dB Coplanar Waveguide Directional Coupler," 22nd European Microwave Conference, DOI: 10.1109/EUMA.1992.335792, September 1992.

[21] A. Cidronali et al. "A MMIC lumped element directional coupler with arbitrary characteristic impedance and its application," 30th European Microwave Conference, Pp: 1-4, DOI: 10.1109/EUMA.2000.338702, October 2000.

[22] Mina Wahib, A. P. Freundorfer, "A miniaturized lumped element directional coupler with parasitics compensation," IEEE International Symposium on Circuits and Systems 2016, May 2016

[23] C. Wang, Y. Li, N. Y. Kim, "A compact 3-dB 90º directional coupler in integrated passive devices manufacturing process for LTE applications," Proceedings of 3rd Asia-Pacific Conference on Antennas and Propagation, Pp.: 1291-1292, DOI: 10.1109/APCAP.2014. 6992756, July 2014

[24] F. Gianesello, C. Durand, D. Gloria, "0.35 dB loss 20 dB coupling directional coupler integrated in 130 nm CMOS SOI technology targeting 3G PA SOC," IEEE 14th Topical Meeting on Silicon Monolithic Integrated Circuits in Rf Systems, PP.: 29-31, DOI: 10.1109/SiRF.2014.6828517, January 2014.

[25] Ying-Cheng Tseng, Tzyh-Ghuang Ma, "On-chip miniaturized 3-dB directional coupler using coupled synthesized CPWs on integrated passive device (IPD) process," 44th European Microwave Conference, Pp.: 81-84, DOI: 10.1109/EuMC.2014.6986374, October 2014.

[26] D. Kajfez, “Scattering matrix of a directional coupler with ideal transformers,” Microwaves, Antennas and Propagation, IEE Proceedings Vol. 146, Issue: 4, pp. 295-297 August 1999.

[27] A. Buonomo and A. L. Schiavo, “Divide-by-Three Injection-Locked Frequency Dividers with Direct Forcing Signal,” Journal of Electrical and Computer Engineering, Volume 2013, Article ID 145314, June 2013.

[28] Chapter 5.8 of [30], page 139ff

[29] Sections 6.2.4 and 6.2.5 of [30], page 165ff

[30] J. Rogers, C. Plett, F. Dai, “Integrated Circuit Design for High-Speed Frequency Synthesis,” Artech House microwave library, ISBN 1-58053-982-3, 2006.

[31] U. Tietze, C. Schenk, E. Gamm, „Halbleiter-Schaltungstechnik,“ chapter 17.3 “Analog-Digital Umsetzer”, P.: 1025ff, ISBN: 978-3-662-48553-8, 16.Auflage, 2019. English version available at Springer: U. Tietze, C. Schenk, E. Gamm, “Electronic Circuits --- Handbook for Design and Applications,” 2nd edition, 2008, ISBN: 978-3-540-78655-9 (eBook)

[32] B. P. Lathi, "Signal Processing & Linear Systems," chapter 3 ”Signal Representation By Fourier Series,” chapter 4 “Continous-Time Signal Analysis: The Fourier Transform,” chapter 5 "Sampling," chapter 8 "Discrete-time Signals and Systems," especially equation (8.21), P.: 558, and chapter 10 “Fourier Analysis of Discrete-Time Signals,” ISBN: 0-941413-35-7, Berkeley Cambridge Press, 1998.

[33] https://commons.wikimedia.org/wiki/File:Bandpass_sampling_depiction.png, accessed 2021-05-11.

[34] Qizheng Gu, “RF System Design of Transceivers for Wireless Communications," chapter 3.4.1 "Basics of Band-pass Sampling," P.: 191, Equations (3.4.3) and (3.4.4), ISBN: 978-0387-24161-6, Springer, 2005.

[35] Y. Chandu, M. Maradi, A. Manjunath and P. Agarwal, "Optimized High Speed Radix-8 FFT Algorithm Implementation on FPGA," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 430-435, doi: 10.1109/ICOEI.2018.8553791, 2018.

[36] https://www.minicircuits.com/pdfs/ZFBDC20-62HP+.pdf, accessed 2021-05-11.

[37] https://assets.nexperia.cn/documents/data-sheet/74AUP1G80.pdf, accessed 2021-05-11.

[38] https://www.analog.com/media/en/technical-documentation/data-sheets/ADF4360-9.pdf, accessed 2021-05-11.

[39] https://www.analog.com/media/en/technical-documentation/data-sheets/229876fa.pdf, accessed 2021-05-11.

[40] https://www.st.com/resource/en/datasheet/stm32f446re.pdf, accessed 2021-05-11.

[41] https://www.st.com/resource/en/reference_manual/dm00135183-stm32f446xx-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf, accessed 2021-05-11.

[42] https://www.st.com/resource/en/application_note/dm00169730-parallel-synchronous-transmission-using-gpio-and-dma-stmicroelectronics.pdf, accessed 2021-05-11.

[43] https://www.st.com/resource/en/user_manual/dm00105823-stm32-nucleo-64-boards-mb1136-stmicroelectronics.pdf, accessed 2021-05-11.

[44] https://www.st.com/resource/en/application_note/dm00046011-using-the-stm32f2-stm32f4-and-stm32f7-series-dma-controller-stmicroelectronics.pdf, accessed 2021-05-11.

[45] https://arm-software.github.io/CMSIS_5/DSP/html/group__groupTransforms.html, accessed 2021-05-11.

[46] https://www.st.com/en/development-tools/truestudio.html, accessed 2021-05-11.

[47] https://www.st.com/en/development-tools/stm32cubemx.html, accessed 2021-05-11.

[48] https://www.mathworks.com/products/matlab.html, accessed 2021-05-11.

[49] https://www.keysight.com/de/de/products/software/pathwave-design-software/pathwave-advanced-design-system.html, accessed 2021-05-11.

[50] https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html, accessed 2021-05-11.

[51] https://www.analog.com/en/design-center/adisimpll.html, accessed 2021-05-11.

[52] https://designers-guide.org/analysis/PLLnoise+jitter.pdf, accessed 2021-05-11.

[53] https://www.planetanalog.com/how-division-impacts-spurs-phase-noise-and-phase/, accessed 2021-05-11.

[54] http://eagle.autodesk.com/eagle/software-versions, accessed 2021-05-11.

[55] https://www.mouser.com/datasheet/2/842/B3742-1092444.pdf, accessed 2021-05-11.

[56] https://assets.nexperia.com/documents/data-sheet/PESD3V3C1BSF.pdf, accessed 2021-05-11.

[57] https://www.onsemi.com/pdf/datasheet/74vcx245-d.pdf, accessed 2021-05-11.

[58] https://www.onsemi.com/pdf/datasheet/NC7WZ17-D.pdf, accessed 2021-05-11.

[59] https://www.onsemi.com/pdf/datasheet/NC7SZ126-D.pdf, accessed 2021-05-11.

[60] https://www.ti.com/lit/ds/symlink/sn74aup1g79.pdf, accessed 2021-05-11.

[61] https://www.ti.com/lit/ds/symlink/sn74aup1g08.pdf, accessed 2021-05-11.

[62] https://www.st.com/resource/en/user_manual/dm00105879-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf, accessed 2021-05-11.

[63] https://www.st.com/en/development-tools/stsw-stm32095.html, accessed 2021-05-11.

[64] https://wiki.analog.com/resources/tools-software/uc-drivers/renesas/adf4360, accessed 2021-05-11.

[65] https://www.visualstudio.com/, accessed 2021-05-11.

[66] https://msdn.microsoft.com/en-us/, accessed 2021-05-11.

[67] A. Rumiantsev, “A Review of VNA Calibration Methods,” EuMIC 2008, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.662.2977&rep=rep1&type=pdf, accessed 2021-05-11.

[68] D. Qiao, Y. Zhao, T. Hung, D. Kimball, M. Li, P. Asbeck, D. Choi, D. Kelly, "Antenna impedance mismatch measurement and correction for adaptive CDMA transceivers," 2005 IEEE MTT-S International Microwave Symposium Digest, Print ISBN: 0-7803-8845-3, June 2005.

[69] J. Buckley, K. G. McCarthy, B. O'Flynn, C. O'Mathuna, "The detuning effects of a wrist-worn antenna and design of a custom antenna measurement system," 2010 European Wireless Technology Conference, September 2010.

[70] V. Kible, R. N. de Lima, K. B. de Brito, A. Bülau and A. Zimmermann, "An UHF Software Defined Reflectometer using Under-Sampling Down-Conversion in the ADC," 2019 49th European Microwave Conference (EuMC), PP.: 523-526, doi: 10.23919/EuMC.2019.8910735, October 2019.

[71] Mohamed Elkeshti, "Implementation and Selection of Algorithms for Optical Ranging Sensors," Bachelor Work, Institute for Micro-Integration at the University of Stuttgart / Hahn-Schickard Stuttgart, November 2020.

[72] K. M. M. Prabhu, "Window Functions and Their Applications in Signal Processing," Taylor & Francis, doi: 10.1201/9781315216386, 2014.

[73] P. K. Dash and S. Hasan, "A Fast Recursive Algorithm for the Estimation of Frequency, Amplitude, and Phase of Noisy Sinusoid," in IEEE Transactions on Industrial Electronics, vol. 58, no. 10, pp. 4847-4856, doi: 10.1109/TIE.2011.2119450, October 2011.

[74] V.N. Ugol'kov, "Methods of Measuring the Phase Shift and Amplitude of Harmonic Signals Using Integral Samples," Measurement Techniques 46, PP.: 495–501, https://doi.org/10.1023/A:1025317616998, May 2003.

[75] P. M. Ramos, A. C. Serra, "A new sine-fitting algorithm for accurate amplitude and phase measurements in two channel acquisition systems," Measurement, Vol. 41, Issue 2, PP.: 135-143, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2006.03.011, February 2008

[76] H. Xiaohong, W. Zhaohua, C. Guoqiang, "New method of estimation of phase, amplitude, and frequency based on all phase FFT spectrum analysis," 2007 International Symposium on Intelligent Signal Processing and Communication Systems, pp. 284-287, doi: 10.1109/ISPACS.2007.4445879, December 2007.

[77] B. Xiong, L. Yang and T. Cao, "A Novel Tuning Method for Impedance Matching Network Based on Linear Fractional Transformation," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 6, pp. 1039-1043, doi: 10.1109/TCSII.2019.2931412, June 2020.

[78] V. M. Zhukov, D. Y. Muromtsev and A. N. Gribkov, "A Computational Control Algorithm for Digital Antenna Matching Device," 2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russia, pp. 596-600, doi: 10.1109/SUMMA50634.2020.9280782, 2020.

[79] **In Appendix A.2:** http://ww1.microchip.com/downloads/en/DeviceDoc/40001842D.pdf, accessed 2021-05-11.

[80] **In Appendix A.2:** http://docs-europe.electrocomponents.com/webdocs/1440/0900766b81440aed.pdf, accessed 2021-05-11.

A Appendix

A.1 ADIsimPLL Outputs for ADF4360-9

The Analog Devices Software "ADIsimPLL" was used to design the PLL subcircuit for the reflectometer. Here the most important outputs of the software are shown.

A.1.1 Schematic

ADIsimPLL outputs the recommended component values corresponding to the settings of the PLL in use. These were taken as base for the choice of near, available component values. Shown are the final design values that were entered into the software to obtain the according simulation results. The resulting schematic is shown in Figure A.1

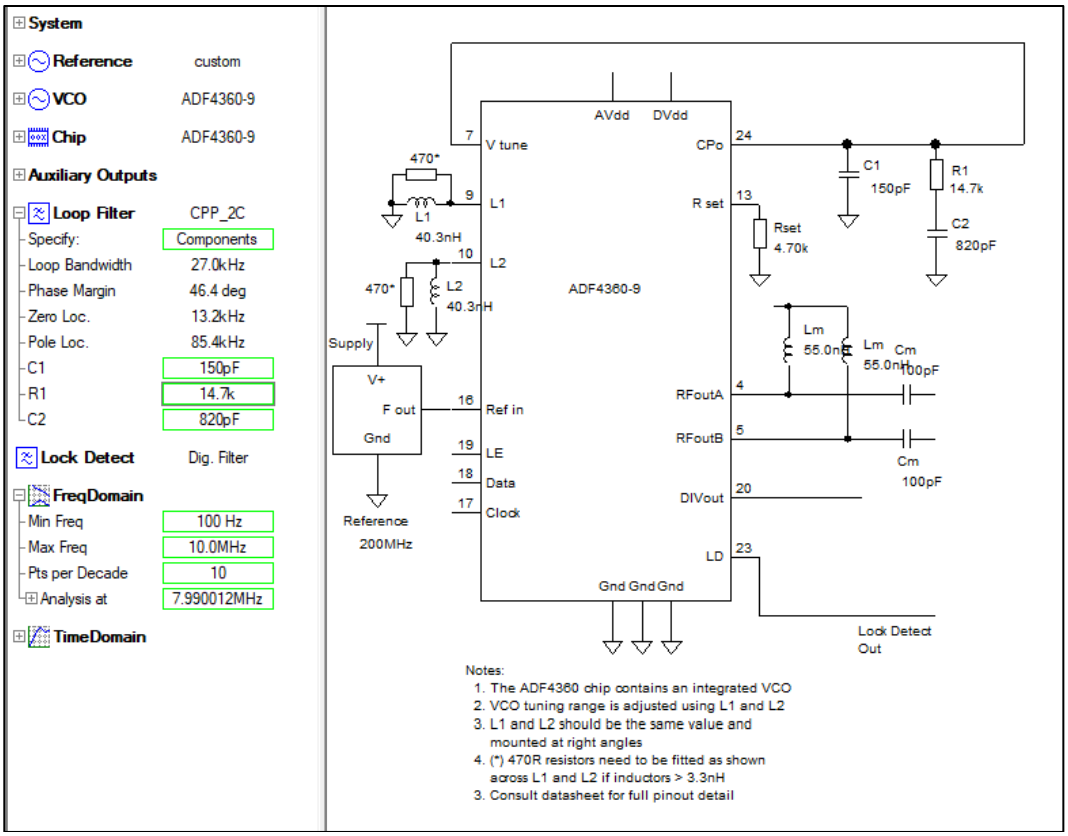

Figure A.1: Schematic of the PLL Circuit

A.1.2 Simulation Results

ADIsimPLL also shows results of time and frequency domain simulations, in Text form and as graphs. First the text output is listed:

```
ADIsimPLL_ADF4360-9.pll analysed at 05/03/21 15:38:59

PLL Chip is ADF4360-9
Notes:
VCO is ADF4360-9
Reference is custom

VCO Divider Outside Loop: division ratio = 32
```

Frequency Domain Analysis of PLL
  Analysis at PLL output frequency of 7.99MHz

Phase Noise Table

Freq	Total	VCO	Ref	Chip	Filter
100	-131.9	-172.2	--	-131.9	-186.9
1.00k	-131.8	-151.6	--	-131.9	-166.9
10.0k	-129.5	-139.2	--	-130.0	-147.0
100k	-143.9	-148.1	--	-146.2	-158.9
1.00M	-159.6	-159.6	--	-185.2	-197.8

Reference Spurious
  Noise and Jitter Calculations include the first 10 ref spurs
  First three spurs: -300 dBc   -300 dBc   -300 dBc

Fractional-N Spur Estimate (worst case)

Phase jitter using brick wall filter
  from 10.0kHz to 100kHz
  Phase Jitter   **0.01 degrees rms**

ACP - Channel 1
  Channel 1 is centred 25.0kHz from carrier with bandwidth 15.0kHz
  Power in channel = **-86.8dBc**

      ----   End of Frequency Domain Results   ----

Transient Analysis of PLL
  Power up transient to frequency of 7.9900125MHz
  Simulation run for 1.00ms

Frequency Locking
  Time to lock to 1.00kHz is 76.2us
  Time to lock to 10.0 Hz is 124us

Phase Locking (VCO Output Phase)
  Time to lock to 10.0 deg is 64.8us
  Time to lock to 1.00 deg is 74.2us

Lock Detect Threshold
  Time to lock detect exceeds 2.50 V is 56.1us

      ----   End of Time Domain Results   ----

The following graphs show frequency domain results (Figure A.2) and time domain results (Figure A.3) of the ADIsimPLL simulations.

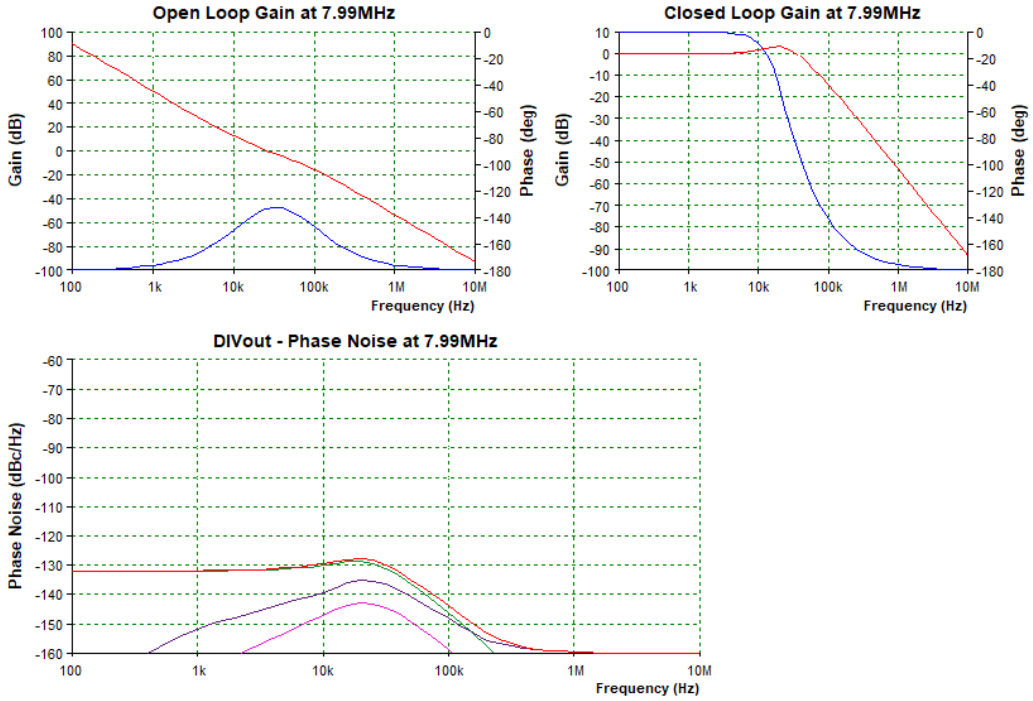

Figure A.2: Frequency Domain Simulation Results

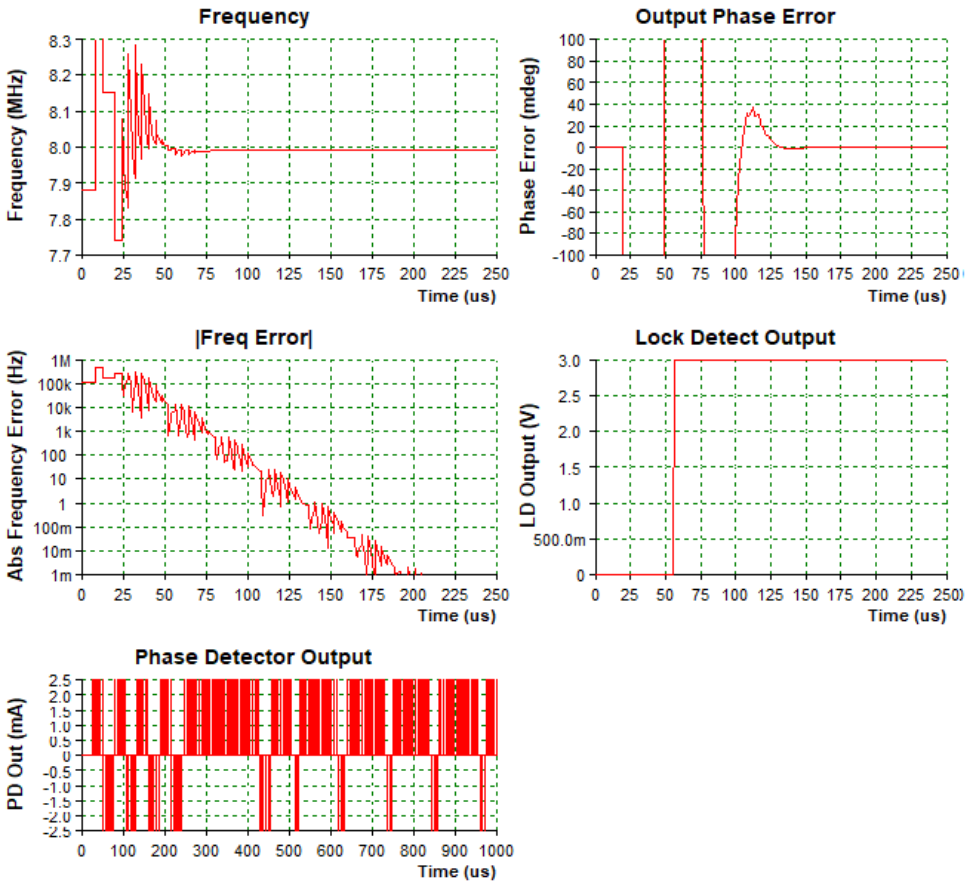

Figure A.3: Time Domain Simulation Results

A.2 Assembly Documentation

The printed circuit boards (PCBs) described in section 5.1 were manufactured and the components and cables were bought. As a next step, the whole system needed to be assembled.

A.2.1 Case

First the cases for two instances of the system were prepared. The purpose of the cases is the protection of the system PCB against external influences, especially electromagnetic irradiation. In this turn, the necessary holes were drilled into the aluminum boxes that had been bought earlier, and one well defined gap was filed into each lid. The result is shown in Figure A.4:

Figure A.4: Cases for the RF Part of the System

A.2.2 Power Supply

In the next step, the power supply of the reflection coefficient measurement system PCB was soldered on both instances and tested. The voltage of the low drop-out (LDO) regulator was only insignificantly different from the desired 3.3V (by some mV). It was important to complete this step before soldering the more expensive integrated circuits such as the LTC2296 ADC [39], to protect the devices from wrong supply voltage and therefore destruction. The PCBs with the power supply populated are shown in Figure A.5:

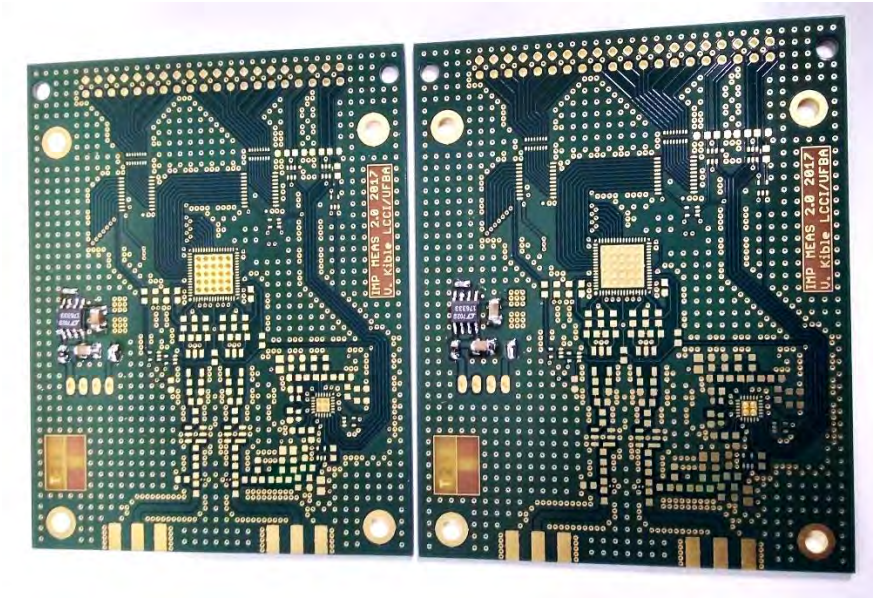

Figure A.5: System PCBs with Power Supply Circuit Populated

A.2.3 Integrated Circuits

The Hahn-Schickard Institute for Micro-Assembly where the soldering took take place has specialized equipment to simplify the task of dispensing the solder-paste onto the solder pads and placing the components correctly: there is a semi-automatic dispenser (Figure A.7) and a manual fine-placer with optical alignment control (Figure A.6). This is especially important for the devices on board that come in QFN packages where the pins are only on the bottom side. The soldering itself was done in a specialized reflow oven.

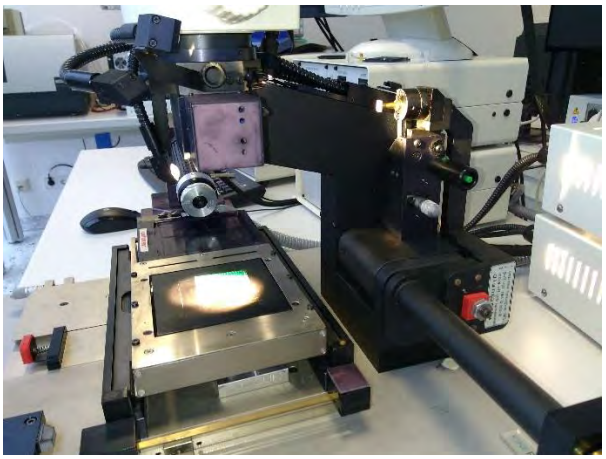

Figure A.6: Manual Fine-Placer

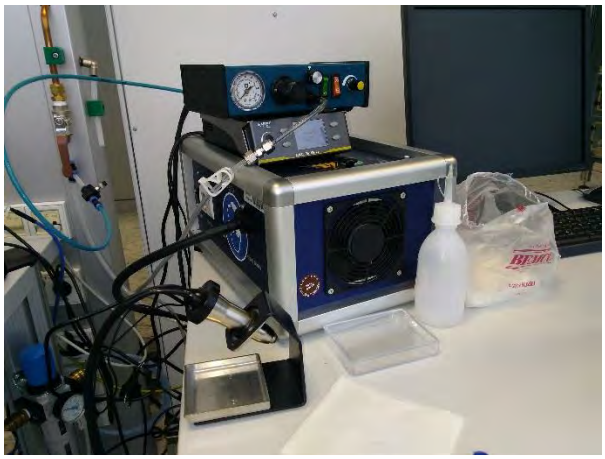

Figure A.7: Semi-Automatic Dispenser

However, the correct amount of solder-paste to be placed on the pads and the correct pattern for placing it had to be found first, so solder tests were necessary. For this purpose, it was decided to buy cheap microcontrollers (PIC18F65K40-I/MR [79]) in the same package as the expensive LTC2296 ADC (the QFN-64 package) and test boards [80]. When everything worked correctly with these test devices, the final components were soldered.

In total, two solder tests had to be completed before the final components could be soldered. The result of the first final solder of the ICs onto the system PCB is shown in Figure A.8.

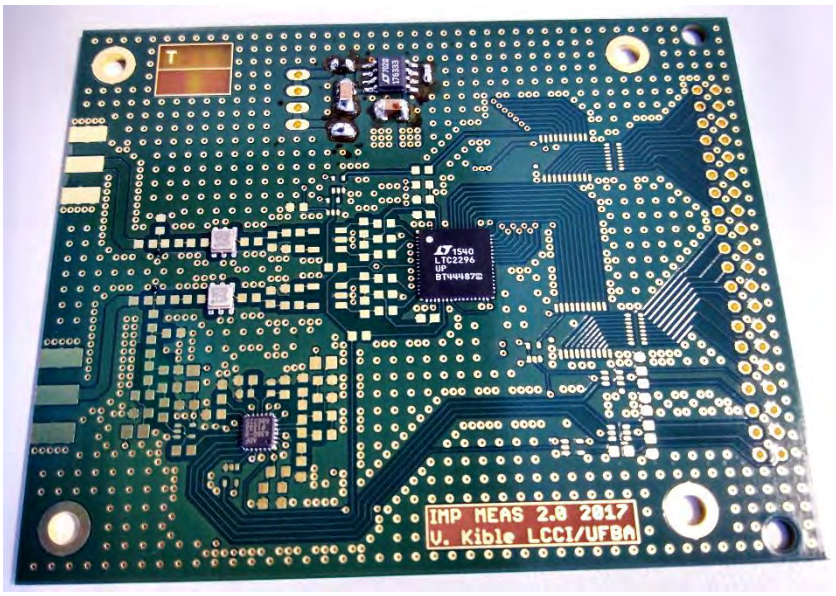

Figure A.8: Result of soldering the ICs onto the first system PCB

It remains to be said that there are still some ICs missing on the board in Figure A.8, but those come in packages that can be soldered by hand more easily and thus were populated later.

A.2.4 Remaining SMD and THT Components

In another step, first the remaining ICs were soldered, then the passive components of the circuit were populated as well as in the very last step the connectors. Because the component values of the PLL loop filter had changed after buying all the components (because of further simulations and calculations), those three components were not populated yet. The resulting status of the PCB is shown in Figure A.9 and Figure A.10.

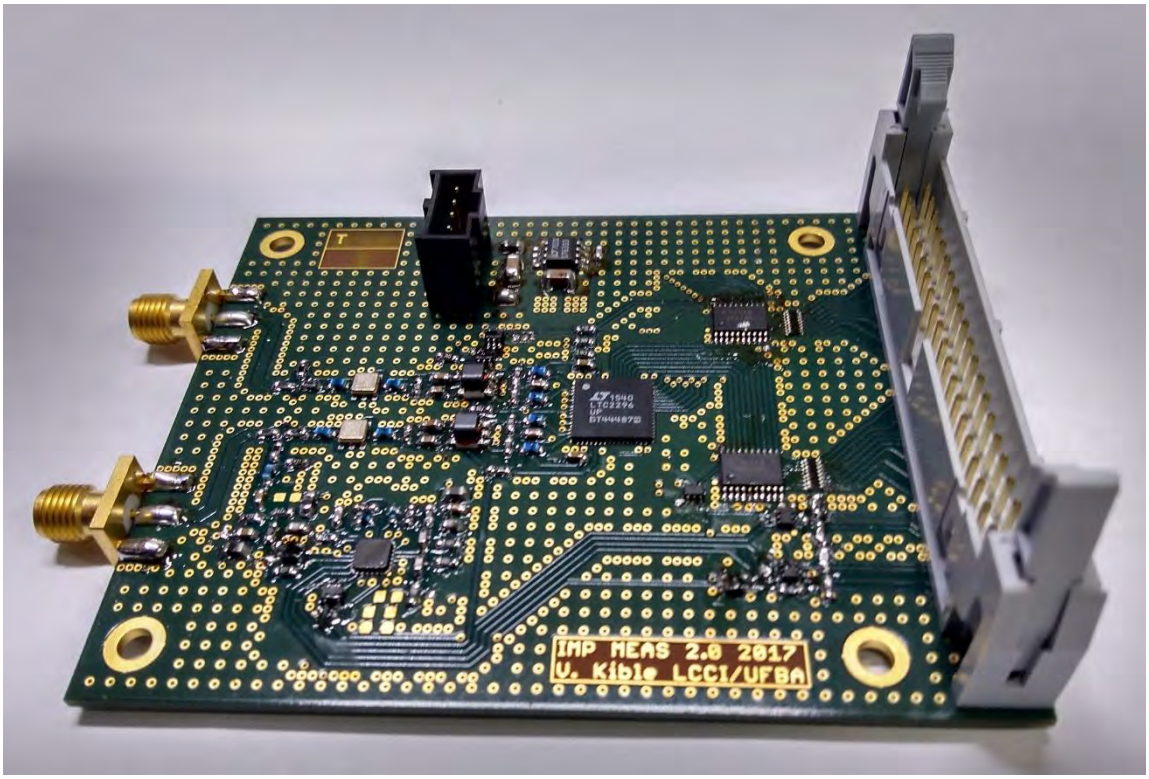

Figure A.9: Almost finished Status of the System PCB (Top Side)

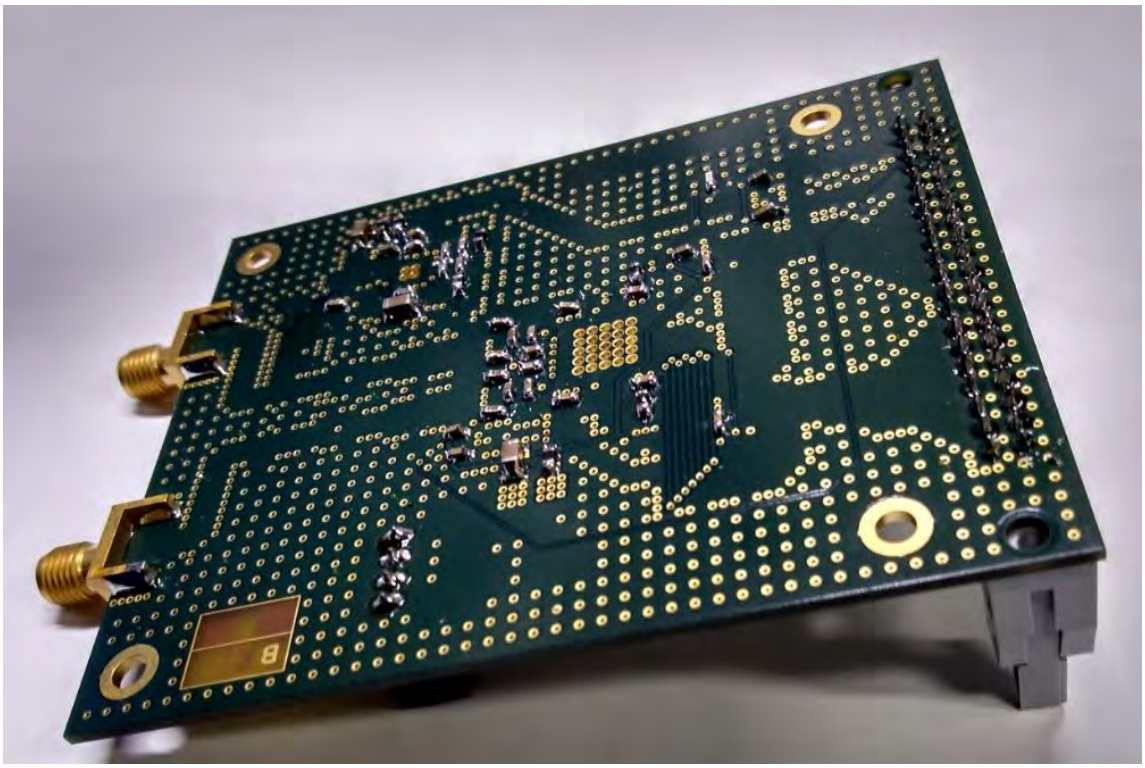

Figure A.10: Almost finished Status of the System PCB (Bottom Side)

A.2.5 Conclusion of Assembly

The remaining components were soldered onto the System PCB and the necessary connectors were applied to the adapter PCB. Also, the ribbon and power cables necessary to connect and power the boards were completed using the respective connectors and ferrites. A photograph of the completed setup can be seen in Figure A.11.

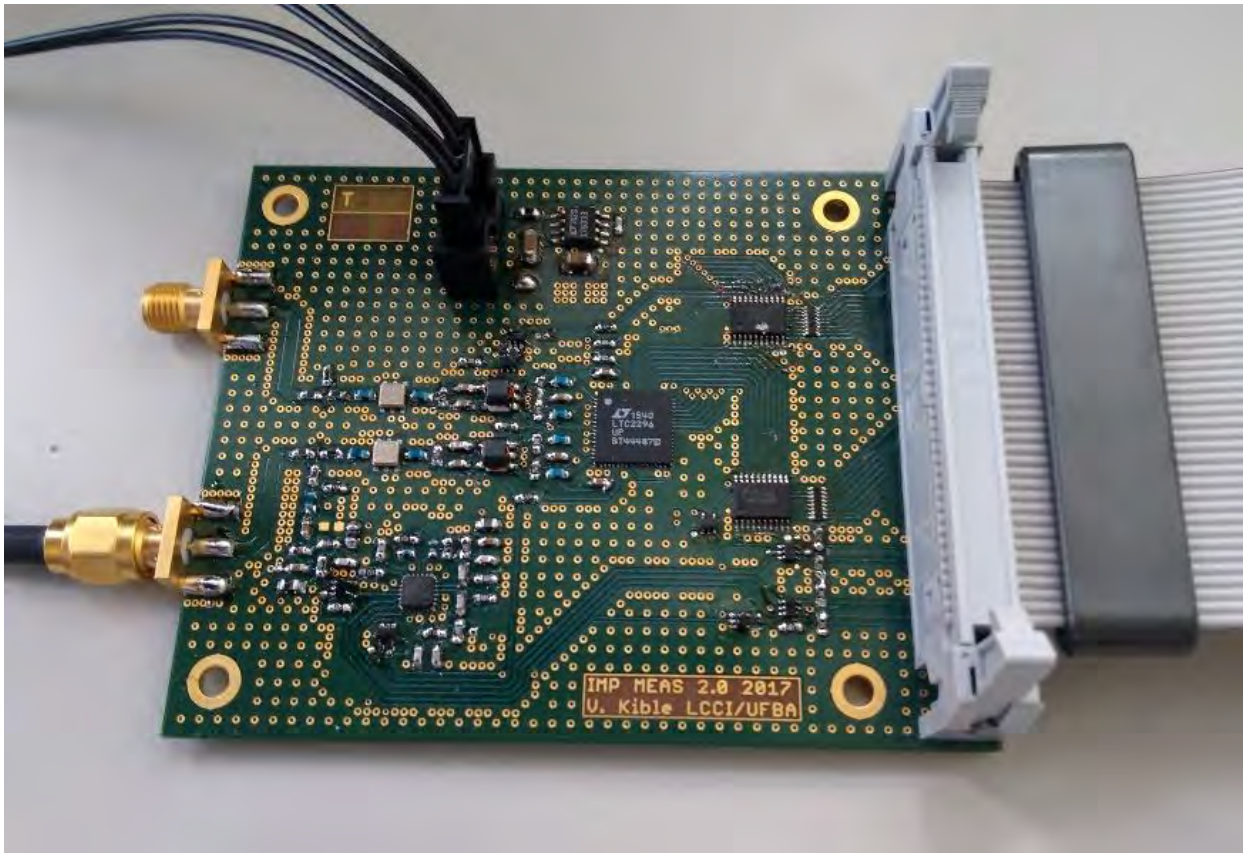

Figure A.11: Completed System PCB (during Test)

When a first test of the assembled system (still not in the case) was attempted, it was found that there were two short circuits, one solder bridge at the supply of the system PCB that was easily removed, and one on the adapter board that connected one of the “no connection”-Pins of the Nucleo board with ground, that was removed by drilling out the pin in the adapter connector.

For higher precision measurements, the system PCB can still be assembled into the case. However, to this point no negative interference was monitored.

A.3 Matlab Script for SOL Correction

```
% Calculation of the Short-Open-Load-correction (SOL) for Reflectometer
% Author:  Volker Kible
% Date: 9.5.2019

clear
clc

syms S11Ar1 S11Ai1 S11Mr1 S11Mi1 S11Ar2 S11Ai2 S11Mr2 S11Mi2
syms S11Ar3 S11Ai3 S11Mr3 S11Mi3 ES ER ED % ESr ESi ERr ERi EDr EDi

S11A1 = S11Ar1 + 1i*S11Ai1;
S11M1 = S11Mr1 + 1i*S11Mi1;
S11A2 = S11Ar2 + 1i*S11Ai2;
S11M2 = S11Mr2 + 1i*S11Mi2;
S11A3 = S11Ar3 + 1i*S11Ai3;
S11M3 = S11Mr3 + 1i*S11Mi3;
%ES   = ESr    + 1i*ESi;
%ER   = ERr    + 1i*ERi;
%ED   = EDr    + 1i*EDi;

EQA1 = S11A1 == 1 / ( ES + ( ER / (S11M1 - ED) ) );
EQA2 = S11A2 == 1 / ( ES + ( ER / (S11M2 - ED) ) );
EQA3 = S11A3 == 1 / ( ES + ( ER / (S11M3 - ED) ) );

S1 = ...
solve([EQA1, EQA2, EQA3], ...
      [ES, ER, ED])

RC_Dat = readtable('RCDat_Values.xlsx');

%%% ES

ES2 = subs(S1.ES, S11Mr1, sym(RC_Dat.S11Mr(1)) ); %8
ES3 = subs(ES2, S11Mi1, sym(RC_Dat.S11Mi(1)) );
ES4 = subs(ES3, S11Ar1, sym(RC_Dat.S11Ar(1)) );
ES5 = subs(ES4, S11Ai1, sym(RC_Dat.S11Ai(1)) );

ES6 = subs(ES5, S11Mr2, sym(RC_Dat.S11Mr(13)) ); %10
ES7 = subs(ES6, S11Mi2, sym(RC_Dat.S11Mi(13)) );
ES8 = subs(ES7, S11Ar2, sym(RC_Dat.S11Ar(13)) );
ES9 = subs(ES8, S11Ai2, sym(RC_Dat.S11Ai(13)) );

ES10 = subs(ES9,  S11Mr3, sym(RC_Dat.S11Mr(17)) ); %12
ES11 = subs(ES10, S11Mi3, sym(RC_Dat.S11Mi(17)) );
ES12 = subs(ES11, S11Ar3, sym(RC_Dat.S11Ar(17)) );
ES13 = subs(ES12, S11Ai3, sym(RC_Dat.S11Ai(17)) );
```

```
%%% ER

ER2 = subs(S1.ER, S11Mr1, sym(RC_Dat.S11Mr(1)) );
ER3 = subs(ER2, S11Mi1, sym(RC_Dat.S11Mi(1)) );
ER4 = subs(ER3, S11Ar1, sym(RC_Dat.S11Ar(1)) );
ER5 = subs(ER4, S11Ai1, sym(RC_Dat.S11Ai(1)) );

ER6 = subs(ER5, S11Mr2, sym(RC_Dat.S11Mr(13)) );
ER7 = subs(ER6, S11Mi2, sym(RC_Dat.S11Mi(13)) );
ER8 = subs(ER7, S11Ar2, sym(RC_Dat.S11Ar(13)) );
ER9 = subs(ER8, S11Ai2, sym(RC_Dat.S11Ai(13)) );

ER10 = subs(ER9,  S11Mr3, sym(RC_Dat.S11Mr(17)) );
ER11 = subs(ER10, S11Mi3, sym(RC_Dat.S11Mi(17)) );
ER12 = subs(ER11, S11Ar3, sym(RC_Dat.S11Ar(17)) );
ER13 = subs(ER12, S11Ai3, sym(RC_Dat.S11Ai(17)) );

%%% ED

ED2 = subs(S1.ED, S11Mr1, sym(RC_Dat.S11Mr(1)) );
ED3 = subs(ED2, S11Mi1, sym(RC_Dat.S11Mi(1)) );
ED4 = subs(ED3, S11Ar1, sym(RC_Dat.S11Ar(1)) );
ED5 = subs(ED4, S11Ai1, sym(RC_Dat.S11Ai(1)) );

ED6 = subs(ED5, S11Mr2, sym(RC_Dat.S11Mr(13)) );
ED7 = subs(ED6, S11Mi2, sym(RC_Dat.S11Mi(13)) );
ED8 = subs(ED7, S11Ar2, sym(RC_Dat.S11Ar(13)) );
ED9 = subs(ED8, S11Ai2, sym(RC_Dat.S11Ai(13)) );

ED10 = subs(ED9,  S11Mr3, sym(RC_Dat.S11Mr(17)) );
ED11 = subs(ED10, S11Mi3, sym(RC_Dat.S11Mi(17)) );
ED12 = subs(ED11, S11Ar3, sym(RC_Dat.S11Ar(17)) );
ED13 = subs(ED12, S11Ai3, sym(RC_Dat.S11Ai(17)) );

format long;
ESd = double(ES13)
ERd = double(ER13)
EDd = double(ED13)

S11A = RC_Dat.S11Ar + 1i.*RC_Dat.S11Ai;
S11M = RC_Dat.S11Mr + 1i.*RC_Dat.S11Mi;

S11_SOLcorr = 1 ./ ( ESd + ( ERd ./ (S11M - EDd) ) )

Cond    = RC_Dat.Condition;
S11Ar   = real(S11A);
S11Ai   = imag(S11A);
S11Mr   = real(S11M);
S11Mi   = imag(S11M);
S11SOLr = real(S11_SOLcorr);
S11SOLi = imag(S11_SOLcorr);

OutputTable = table(Cond,S11Ar,S11Ai,S11Mr,S11Mi,S11SOLr,S11SOLi);

writetable(OutputTable,'SOL_corrected2.xlsx');
```

A.4 Publications

A.4.1 As Author

I. V. Kible, K. B. de Brito and R. N. de Lima, "Quadrature frontend with directional coupler for RF reflection coefficient measurements," 2017 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), 2017, pp. 1-5, doi: 10.1109/IMOC.2017.8121102.

II. V. Kible, R. N. de Lima, K. B. de Brito, A. Bülau and A. Zimmermann, "An UHF Software Defined Reflectometer using Under-Sampling Down-Conversion in the ADC," 2019 49th European Microwave Conference (EuMC), 2019, pp. 523-526, doi: 10.23919/EuMC.2019.8910735.

III. V. Kible, R. N. de Lima, K. B. de Brito, A. Bülau and A. Zimmermann, "Quadrature Block for UHF Reflection Coefficient Measurements Using a Directional Coupler and Injection Locking," in IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 1, pp. 275-285, Jan. 2020, doi: 10.1109/TIM.2019.2895480.

A.4.2 As Co-Author

IV. K. B. Brito, R. N. de Lima, V. Kible and R. C. S. Freire, "A 2.45 GHz CMOS active quasi-circulator with a built-in rectifier," 2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS), 2018, pp. 1-4, doi: 10.1109/LASCAS.2018.8399910.