Bachelor- Thesis

Evaluation and implementation of a signal chain for low-field magnetic resonance imaging

by
Ole Oggesen
Student- ID: 378883

Supervisor:
Prof. Dr.-Ing. Olaf Hellwich
Prof. Dr. Tobias Schäffter
Dr. Lukas Winter

Technische Universität Berlin
Fakultät IV
Institut für technische Informatik und Mikroelektronik
Fachgebiet Computer Vision and Remote Sensing
Berlin, 20. Juli 2021

Abstract

This work implements and evaluates the signal chain for low-field magnetic resonance imaging. The hardware modules used are cost-efficient and mostly consisting of Open Source Hardware, which is controlled by Open Source software. Such technology could enable more portable, and more cost-efficient MR technology with new application areas and therefore easier access to MRI.

The signal chain consists of the transmission and amplification of radio frequency (RF) pulses and the generation and amplification of currents used by the gradient coils of an MR system.

In order to evaluate the performance of the signal chain, each module was investigated individually by measuring and evaluating the in- and output signals. A systematic evaluation was performed in Python and parameters such as pulse shapes, amplitudes and timings were investigated. The evaluation is performed for a reference desktop MR system at B0 = 0.01T (f=4.3576MHz), but in principle can be scaled to other low-field MR systems. The results were evaluated, discussed and alternative modules presented. All Measurements and the evaluation software are available under the following link: `https://github.com/ooggesen/BA`.

Zusammenfassung

In dieser Arbeit wurde die Signalkette für den Aufbau eines Niedrigfeld MR Tomographen implementiert und evaluiert. Die eingesetzten Hardware Module sind kostengünstig und überwiegend Open Source Hardware und werden mit Open Source Software angesteuert. Dies könnte die Anwendung portabler, kostengünstiger MR Technologie ermöglichen, wodurch unter anderem neue Anwendungsgebiete und ein einfacherer Zugang zur Technologie entstehen.

Die Signalkette beinhaltet zum einen das Aussenden und die Verstärkung von Hochfrequenz (HF) Pulsen und zum anderen die Generierung und Verstärkung von Strömen für die Gradientenspulen.

Um die Spezifikationen dieser Signalkette bewerten zu können, wurden die Ein- und Ausgangssignale jedes Moduls einzeln untersucht und ausgewertet. Zur systematischen Auswertung wurde ein Programm in Python geschrieben, welches alle Metriken aus den Messaufnahmen auswertet. Dabei wurden die Form, Amplitude und das Timing der einzelnen Signale untersucht. Die Leistungsauswertung der Module erfolgte für einen Referenztomographen bei B0=0.01T (f=4.3576MHz), kann aber im Prinzip auch auf andere Niedrigfeld MRTs angewendet werden. Die Ergebnisse werden evaluiert, diskutiert und Alternativen zu jedem Modul vorgestellt.

Alle Messungen und das Evaluationsprogramm sind unter dem folgenden Link erreichbar: `https://github.com/ooggesen/BA`

Contents

Eidestattliche Versicherung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den July 20, 2021

____

Ole Oggesen

1 Introduction

Magnetic resonance imaging (MRI) is a very powerful tool for clinical imaging, giving insight into soft tissue properties and metabolic processes [1]. Technological improvements have led to increased availability and functionality of this technology. However due to its high cost in installation and maintenance and the complexity of MRI systems, its distribution is worldwide extremely inhomogeneous [2]. Given this imbalance, there is a strong need in revisiting the approach of using low field (0.05 -0.3 T [3]) MRI systems, in order to improve accessibility and enable point-of-care imaging. Even though these systems do not provide the signal to noise ratio (SNR) of their high field (B_0>1.0T) counterparts, they nevertheless deliver diagnostically useful information that is highly beneficial for many clinical applications [3].

In addition low field MRI scanners have substantial advantages in comparison to their high field counterparts [3]. The magnets can be build relatively lightweight e.g. using permanent magnets in Halbach arrays, which at the same time are cost efficient [3]. This might result in the possibility of a portable scanner implementation. Furthermore is the static magnetic field B_0 much lower and therefore much safer to operate [2]. Magnetic materials inside the patients body (e.g. implants) and outside the scanner (e.g. other medical equipment) experience much lower attraction forces and pose a lower risk to the patient and operator. At the same time the specific absorption rate (SAR) is much lower for a given input power. Implants experience not only lower RF induced heating at low-fields, they can be better imaged due to lower susceptibility artifacts [2]. For all these and many more reasons it is scientifically and clinically attractive to investigate low-field MRI even further. To improve accessibility of MRI globally and create widespread impact, MR scanner costs need to be reduced substantially [3]. Low-field MR hardware can be implemented with more cost-efficient technologies. At higher fields the complexity of magnet designs increases rapidly, electronics are more challenging to design due to higher sampling or increasing RF losses with increasing frequencies.

Another key component to reduce costs is by applying open source hardware practices

[4]. It has been shown that open source scientific hardware of similar functionality can be built by investing only 10% of the cost of proprietary hardware [4]. More recently, several open source hardware projects have been launched, which can be used to construct a low-field MRI scanner.

In this work the signal chain for a potential low-field MRI scanner is implemented, which is compiled from these open source hardware and software parts. The signal chain contains a RF power amplifier (RFPA), a gradient amplifier and a transmit-receive (TR) switch with a low-noise pre-amplifier. Each module is evaluated in test-measurements at B_0=0.1T. Together with the introduction of the evaluated parts, basic requirements for each parts, as well as alternatives are presented, to give an overview over other possible implementations.

2 Theory

In this chapter the basic principles and practical concepts of MR imaging are presented and explained. This includes the physical process consisting of the behavior of the spin of an atom in a static magnetic field, its response to an excitation of a perpendicular oscillating field and how this contains information about the specific tissues or materials.

After a general overview of the theoretical physics behind MRI, the necessary hardware to apply this physics practically is introduced briefly.

2.1 Fundamental principles of MRI

The basic principle of MRI lies in the interaction of the spin of the hydrogen (H) nuclei with an external magnetic field [1]. When hydrogen is introduced to a magnetic field, the spin states are alternated. For many hydrogen nuclei (e.g. in a water sample) these alterations of the spin states lead to a measurable magnetization of the sample, which is aligned along the direction of B_0 and precesses around it [1].

The movement of the magnetization of the nuclei can be understood in analogy to a spinning gyroscope, which is electrically charged. This minuscule charge, creates a magnetic dipol moment $\vec{\mu}$, which will tend to align with the external field B_0. In this process, the magnetic dipol moment precesses just like a gyroscope around the external field until it spins in an equilibrium state around the B_0 axis. The precession frequency called Larmor frequency scales with the field strength of the external field (2.1) [1].

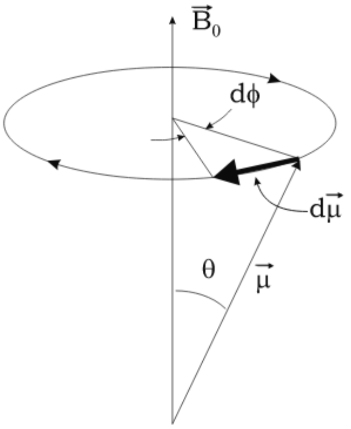


Figure 2.1: Visualization of a spinning magnetization[1, page 3]

$$f_{Larmor} = \gamma \cdot \mathrm{B}_0 \tag{2.1}$$

With $\gamma \approx 42.577\ \frac{\mathrm{MHz}}{\mathrm{T}}$.

This magnetiztion vector of the sample can be rotated into the plane perpendicular to B_0 by applying a radio frequency (RF) pulse at Larmor frequency, which is also oriented perpendicular to the external field B_0 [1]. This is visualized by the flip angle θ in the figure 2.1. After the RF pulse ends, the magnetization starts to align again with the external magnetic field. The time needed for 63 % of this realignment is called T_1 relaxation time. It is tissue dependent and the speed in which it aligns gives information about the composition of the physical object that is being imaged [1]. Another contrast parameter, is the so called T_2 relaxation, which captures the decay in magnetization in the plane perpendicular to B_0. The magnetization in this plane is generating the MR signal and T_2 is the time for the magnetization to fall to about 37% of its peak value and it is much shorter (typically 100-300ms) then T_1 (0.5-5s) [1].

If a spatially changing magnetic field (B_{grad}) is applied in parallel to B_0 over the sample, the Larmor frequency is manipulated(Eq. 2.2) [1]. This magnetic field component is designed to apply a linear magnetic field gradient over the B_0 field and it is therefore referred to as gradient. With three perpendicular gradient magnetic fields, the external magnetic field B_0 can be adjusted at every point in the sample. This allows selective excitation of the sample, if the gradient field is generated along side the transmission of the RF pulse. If the gradients are played out simultaneously to the measurement of RF signals, the spatially varying Larmor frequency creates frequency encoding over the sample. When the gradient is generated independently from those two cases, it leads to spatially varying dephasation. This procedure leads to a variety of pulse sequence designs, which are however not further discussed in this work. The frequency and phase data, often referred to as k-space, can than be evaluated by a Fourier Transform to extract the information and e.g. create an image [1].

$$f_{Larmor}(z,t) = \gamma \cdot (B_0 + z \cdot B_{grad}(t)) \tag{2.2}$$

Due to inhomogeneities in the external magnetic field B_0, the magnetization flipped into the plane perpendicular to B_0 dephases locally [1]. This dephasing of the magnetization leads to a lower measured signal strengths and requires fast sampling of the signal after the initial RF pulse.

To receive the information, which is lost by the fast decay, based on dephasing, an additional RF pulse can rephase the magnetization resulting in a short measurable peak at the echo time (T_E) [1]. This is the so called spin echo (SE) imaging technique, which in its simplest form applies two RF pulses. The first applying a $\frac{\pi}{2}$ flip angle and the second applying a π, which effectively flips the magnetization in the x-y-plane and causes rephasing. The time between the two RF pulses (τ) define the time at which the SE occurs (Eq. (2.3)) [1].

$$T_E = 2 \cdot \tau \tag{2.3}$$

The detection of the RF signal created by the precessing magnetic dipol moment, as well as the generation of the RF pulse, can be done with the same coil, which must be tuned to the individual Larmor frequency [1].

2.2 Signal chain of a MRI system

Basic hardware components needed for an MR imaging experiment are schematically illustrated in Figure 2.2. In this chapter the hardware needed to produce, transmit and receive signals that are used to create a basic MR imaging experiment are introduced.

For the process of MR imaging to succeed, a scanner needs a console to create RF and gradient pulses and to sample the received MR signal. In addition a gradient power amplifier (GPA) and gradient coil to create a spatially varying magnetic field over the sample, as well as a RFPAand RF coil to create the RF pulse needed to tilt the orientation of the magnetization.

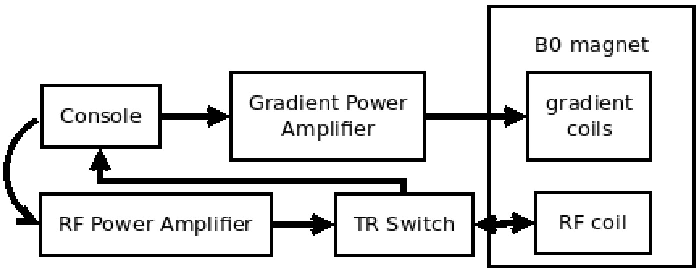


Figure 2.2: Illustration of a possible scanner setup.

Each part in the system setup has specific requirements and sets different limitations, which will shape the imaging performance of the overall system.

Console

The console controls the the timing, amplitude, shape and frequency of the pulses played out in the appropriate coils. It also serves as an interface for an operator and therefore is used to control the whole scanner as well as deliver feedback. In this case it is also capable of creating and receiving RF pulses on its own.

The requirements to the console in the transmission side:

- A sufficiently high sampling rate, to transmit RF pulses at the desired Larmor frequency
- At least one RF and three (x, y, z)-gradient gradient output channels
- Digital I/O channels for gating the RF power amplifiers, switching (e.g. T/R switches) and other purposes (e.g. controlling safety monitoring hardware)
- Synchronization between RF and gradient pulses and I/O channels
- High output sensitivity, which is achieved through high resolution digital-to-analog converters (DACs)

The requirements on the reception side are:

- Amplification of the received MR signal, which is in the order of μV, at low noise
- Sampling rate sufficiently high enough to sample RF signals at the desired Larmor frequency
- At least one but preferably multiple receive channels
- Synchronization between transmitted and received signal
- High input sensistivity which is achieved with high resolution analog-to-digital converters (ADCs)

RF power amplifier

To achieve a desired flip angle, by using short RF pulses, the output of the console ($<$ 1V) needs to be amplified using RF power amplifiers(Eq. (2.4)).

If a rectangular RF pulse (with amplitude B_1 in the sample) is perpendicular to the static magnetic field B_0 and persists for t=τ, then the change in the flip angle ($\Delta\theta$) is defined by equation (2.4) [1].

$$\Delta\theta = \gamma B_1 \tau \tag{2.4}$$

The flip angle can be modified by either changing B1 amplitude or the RF pulse duration. If the RF pulses are made longer, then the minimum TE/TR also needs to be increased, which has implications on the imaging speed and contrast. The requirements for the RFPA are:

- Linearity at the investigated frequency, which guarantees that an input voltage is reliably amplified to a defined output voltage
- Sufficient amplification or gain, depending on the desired application and RF coil transmit efficiency
- Gating circuit, which enables amplification only during transmission

Transmit-receive (TR) switch

The TR switch regulates the power flow of the RF signals. This allows the usage of the same coil for transmission (TX) as well as reception (RX). In the TX scenario the high voltage RF pulse from the RFPA is guided to the RF coil. In the RX scenario the low voltage RF MR signal is routed from the RF coil to the console.

The main requirements to the TR-switch are:

- Good isolation between the TX and RX path [5]
- Sufficiently short switching times [6]

The TX to RX isolation directly limits the amount of power which can get transmitted without damaging the pre-amplifier in the receive chain, which is adjusted to amplify the received MR signal at very low amplitudes. In the case of this MRI scanner, it sets an upper limit of 1W transmit power, but in high field systems peak powers of 10-20 kW are common [7].

The switching time, define the minimum time needed, to switch the TR switch from the TX to RX scenario and therefore the minimum T_E .

Gradient power amplifier (GPA)

The gradient pulses are transmitted by the console typically at low voltage levels and need to be amplified by a gradient power amplifier in order to generate sufficiently strong gradient fields. This creates a gradient magnetic field over the sample.

Main requirements for the GPA are:

- At least three independent output channels
- Sufficient output power
- Sufficiently fast in reaching its maximum amplitude (slew rate)

Gradients are typically switched at frequencies in the lower kHz range and commercial whole body systems operate GPA with >100A per channel. Smaller MR systems have higher gradient coil efficiencies ($\frac{\mathrm{mT}}{\mathrm{mA}}$) and can be operated with lower output currents, while providing the same gradient strengths [2].

3 Methods

This chapter describes the functionality of each module in the measurement setup and its configurations as well as a description of the experimental approach and laboratory equipment. Finally, the evaluation procedure is described and links to the hardware and software repositories are presented.

3.1 Hardware

This chapter describes the hardware modules that have been selected and evaluated as a potential signal chain for a low-field MRI scanner at B_0 ≈0.1 T (f=4.3576MHz) [8].

3.1.1 Console

Usual MRI consoles are relatively costly due to customized application specific design and small quantity production [3]. But since many of the MRI specific RF signal chain components and signal processing modules can be implemented with a modern FPGA, modern software defined radios could be used as an alternative [3]. At a relatively low price of ≈ 500€ comes the Red Pitaya (RP). It has been demonstrated that the RP could be utilized as a fully functional console named Open Source Console for Real-time Acquisition (OCRA) in tabletop MRI [9]. In our measurements the Red Pitaya 122 is utilized.

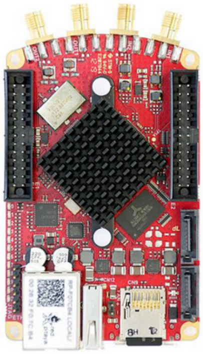


Figure 3.1: Photograph of the RP [10]

	RF inputs	**RF outputs**
Channels	2	2
Input/ Load resistance	50Ω	50Ω
Sampling rate	$122.88\frac{MS}{s}$	$122.88\frac{MS}{s}$
ADC resolution	16 bits	
DAC resolution		14 bits
Full scale voltage range	±0.25V	±0.5V
Bandwidth	300kHz – 550MHz	300kHz – 60MHz

	extension connectors
digital IOs	16
analog inputs	4 channels, 0-3.5V, 12 bit
analog outputs	4 channels, 0-1.8V, 12 bit
communication interfaces	I2C, UART, SPI
available voltages	+5V, +3.3V, -4V

Processor	DUAL CORE ARMCORTEX A9
FPGA	Xilinx Zynq 7020 SOC

Table 3.1: Specifications Red Pitaya 122-16 [10]

The RP functions as arbitrary waveform generator, which can be used as a spectrometer to play out RF waveforms for MRI with a 14 bits DAC resolution (Tab. 3.1). Its sampling rate is sufficient for all low-field applications ($B_0 \leq 0.5$T) [3]. The digital IO pins can be used to generate a gate pulse, which activates the RF signal path and the GPA can be controlled via an SPI interface. Finally it should also be able to measure the MR signal with 16 bits DAC resolution.

In OCRA and in a similar system called MaRCoS, the RP is controlled via a Server Client Model, which communicates via the RJ45 (Ethernet) Port with a local machine. This allows the control of the RP with a Python program at the local machine.

The RP works as a multiprocessing unit, with the FPGA and CPU working independently from one another. Both processors can read and write from the same shared RAM. This has serveral advantages. The CPU can be programmed in a high level language like C and the FPGA provides very high accurate timing and hardware acceleration [9]. This allows the possibilty of realtime control over the scanner, while it operates on a patient or phantom [9]. The figure 3.2 present more information about the implementation of the server console.

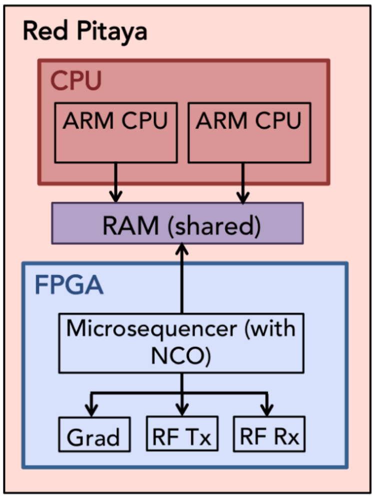


Figure 3.2: Architecture of the RP System [9, page 38]

3.1.2 RF power amplifier

The RFPA amplifies the RF signals from the console to achieve the desired flip angle. The RFPA (fig. 3.3) used in this work is a class A amplifier based on BLF188XRU transistor with a similar architecture as in Cosi transmit [11]. It consists of a driving stage and an amplification stage which results in a total of $\approx 54.3dB$ gain over a frequency range of 500 kHz - 12 MHz. In order to limit the maximum amplification of the RFPA based on the output of the RP, two attenuators are used. One with a fixed attenuation of 30dB (ATT 1,5-120) and the other scalable from 0-110 dB(50DR-001).

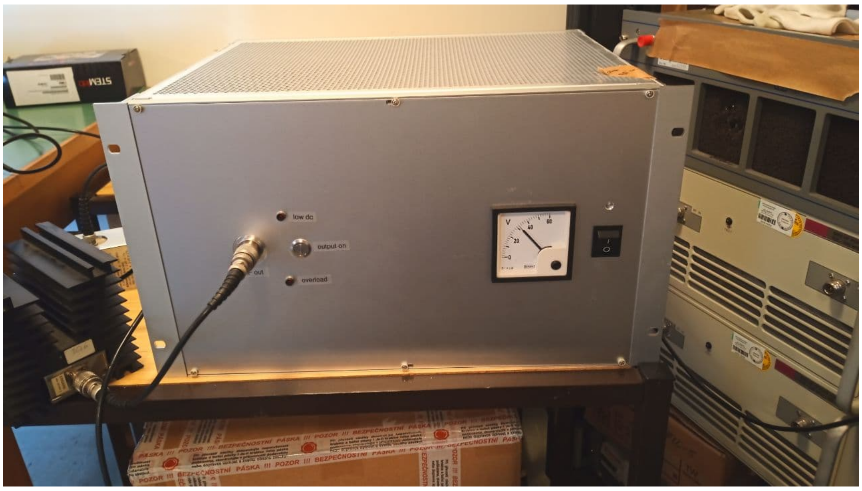


Figure 3.3: Photograph of the RFPA

Gain@4MHz	54.3 dB
Bandwith	500kHz - 12MHz
Enable	5V 1%
Max load power	1kW
Max duty cylce	10%
Max pulse duration	20 ms
R_{out}@4MHz	48.2 Ω
R_{in}	50 Ω

Table 3.2: Specifications of the RFPA

The RFPA used in the measurements, is an updated version of the one published with cosi-transmit, which you will find under the link below. This current version will soon be made available.

`https://github.com/opensourceimaging/cosi-transmit/tree/master/RF%20Power%20Amplifier%20(RFPA)`

3.1.3 TR-switch

The TR-switch regulates the power flow of the RF signals between the RF coil and the MRI console. It routes the high TX signal (typically 10-100V) from the RFPA to the RF coil and the received signal (μV) from the RF coil to a pre-amplification before it is digitized by the ADC from the RP. In this work the TR Switch from the university of Dortmund (3.4) is used and adjusted to our operating frequency of 4.3576MHz, by changing the capacitance (C=740pF) and inductance (L=2.7μH) values of the pi-network in the design.

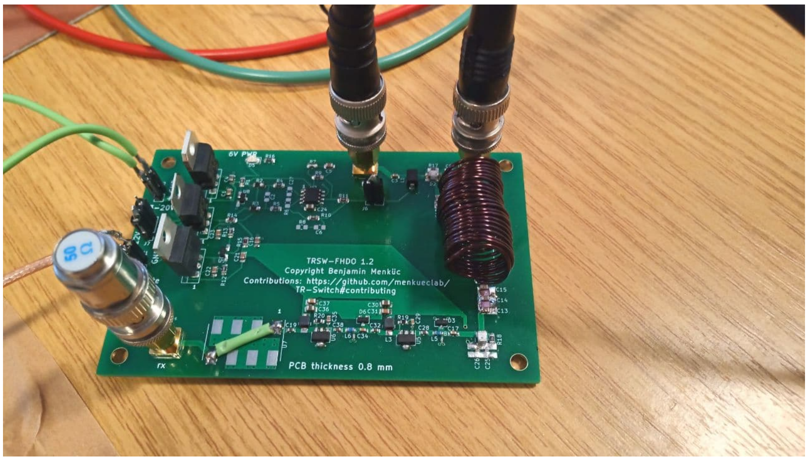

6V PWR
TRSW-FHDO 1.2
Copyright Benjamin Menküc
Contributions: https://github.com/menkueclab/
TR-Switch#contributing
PCB thickness 0.8 mm
rx

Figure 3.4: Photograph of the TR-switch

The design uses MA4P7001F-1072T PIN diodes, which are controlled via the gate pulse at the tx-gate connector. If 5 V are applied to the tx-gate connector, then the connection between TX and the RF Coil is at low impedance, while the TX path is isolated from the RX path. This should be the case if RF signals are being sent to the RF coil. If 0 V are applied, then the RX and the RF coil connectors are conducting. This gate pulse is generated by the RP and in our experiments functions as a trigger for the RFPA as well as the TR-switch. The circuit has two integrated low noise (1.2dB) preamplifiers (PHA-13LN+, ≈ 48.8dB) to amplify the received signal.

In addition to the isolation by the pi-network, the design implements protective diodes (BAV99) prior to the PHA-13LN+ to protect the RX path from critical amplitudes

The TR-switch design used can be found under the link below.

`https://github.com/menkueclab/TR-Switch`

3.1.4 GPA-FHDO

The GPA-FHDO (fig. 3.5) is connected via a RJ45 (Ethernet connector) to an interface sitting on the I/O pins of the RP and it has four independent channels.

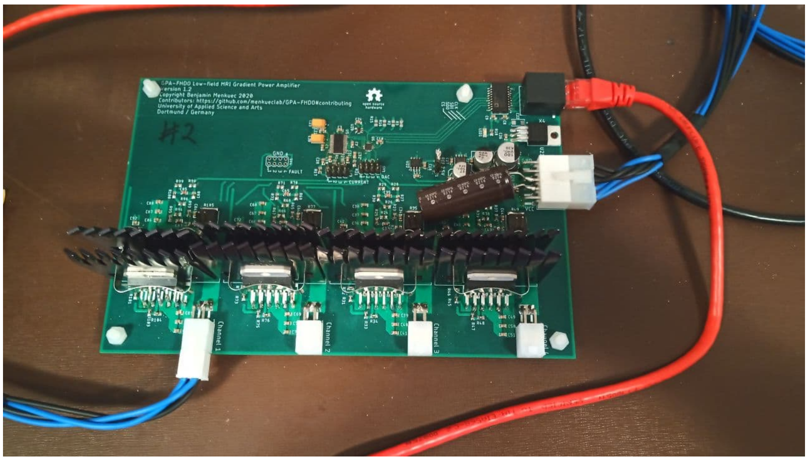


Figure 3.5: Photograph of the GPA-FHDO

The output current range of the GPA-FHDO is adjustable. In this case the maximum current output swing is set to ±6 A, but up to ±10 A is possible. The current is delivered by a push-pull-Howland current source, which itself is controlled by a voltage output from a DAC. The current range can be controlled by evenly distributed values over the whole DAC range [12].

The output current is measured through a 200mΩ shunt resistor, which is connected to a 2.5V offset voltage, to allow measurements of bidirectional currents [12]. This makes a calibration of the output current possible, to counteract temperature and gradient coil changes. A calibration of all four channels only takes about 13 ms. This promises high linearity.

parameter	GPA-FHDO
No of channels	4
Interface	Single-channel-SPI
DAC resolution	16 bits
Supply voltages	13 – 25V
Max. current output per channel	±10A
Peripherals	ADC ADS8684 for monitoring and insequence calibration
Rise-Time	32μs
Temperature offset error	Max. 1%

Table 3.3: Specifications of the GPA-FHDO [12]

In the measurements, the GPA-FHDO is powered by the Meanwell UHP-750-24 power supply with a supply voltage of 24 V. To simulate a similar load at the current outputs as in the finished scanner, a dummy load is connected to the individual output. It is compiled of a coil with similar inductance ($\approx 50\mu$H, 1 Ω) than the gradient coils, and if current information is needed a 0.1Ω (±5%) resistor is added in series. The voltage over the resistor is then measured and transformed to current information via Ohms law.

The GPA-FHDO design and the interface to the RP can be found under the following links.

`https://github.com/menkueclab/GPA-FHDO`
`https://github.com/menkueclab/GPA-RP-Adapter`

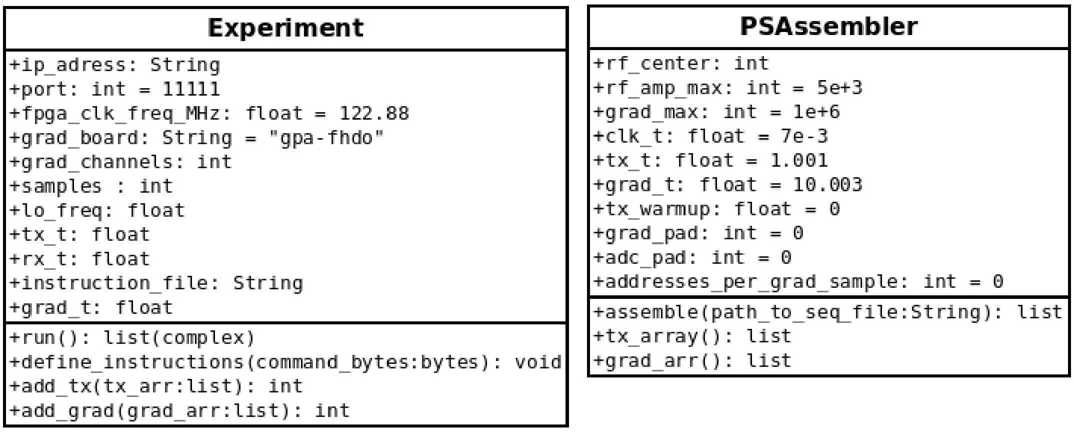
3.2 Operating software

In order to operate the signal chain and play out custom imaging techniques for MRI, two different software approaches are available. Introduced together with the first MRI scanner, based on the RP, named OCRA software framework [13], which is an assembly like language defining the exact timing and shape of the RF and gradient pulses.

Another approach, in programming pulse sequences for MRI, is Pulseq [14], which is a vendor agnostic, low level language that defines hardware events. Subsequentially individual compilers translate the instructions to different proprietary and open source MR scanners [14][15].

Both approaches can be implemented in Phyton and and run on the RP.

The figure 3.6 illustrates two important classes for a successful implementation of OCRA and Pulseq on the RP.



Experiment

```
+ip_adress: String
+port: int = 11111
+fpga_clk_freq_MHz: float = 122.88
+grad_board: String = "gpa-fhdo"
+grad_channels: int
+samples : int
+lo_freq: float
+tx_t: float
+rx_t: float
+instruction_file: String
+grad_t: float
```

```
+run(): list(complex)
+define_instructions(command_bytes:bytes): void
+add_tx(tx_arr:list): int
+add_grad(grad_arr:list): int
```

PSAssembler

```
+rf_center: int
+rf_amp_max: int = 5e+3
+grad_max: int = 1e+6
+clk_t: float = 7e-3
+tx_t: float = 1.001
+grad_t: float = 10.003
+tx_warmup: float = 0
+grad_pad: int = 0
+adc_pad: int = 0
+addresses_per_grad_sample: int = 0
```

```
+assemble(path_to_seq_file:String): list
+tx_array(): list
+grad_arr(): list
```

Figure 3.6: Class diagram of PSAssembler [16] and Experiment [17]

The Experiment class acts as an interface to the RP and has all communications and input variables implemented which are necessary to define the whole pulse sequence. It is part of the server-client model which controls the RP [9].

The PSAssembler class is an interpreter of Pulseq instructions and translates them into data understandable for the experiment class.

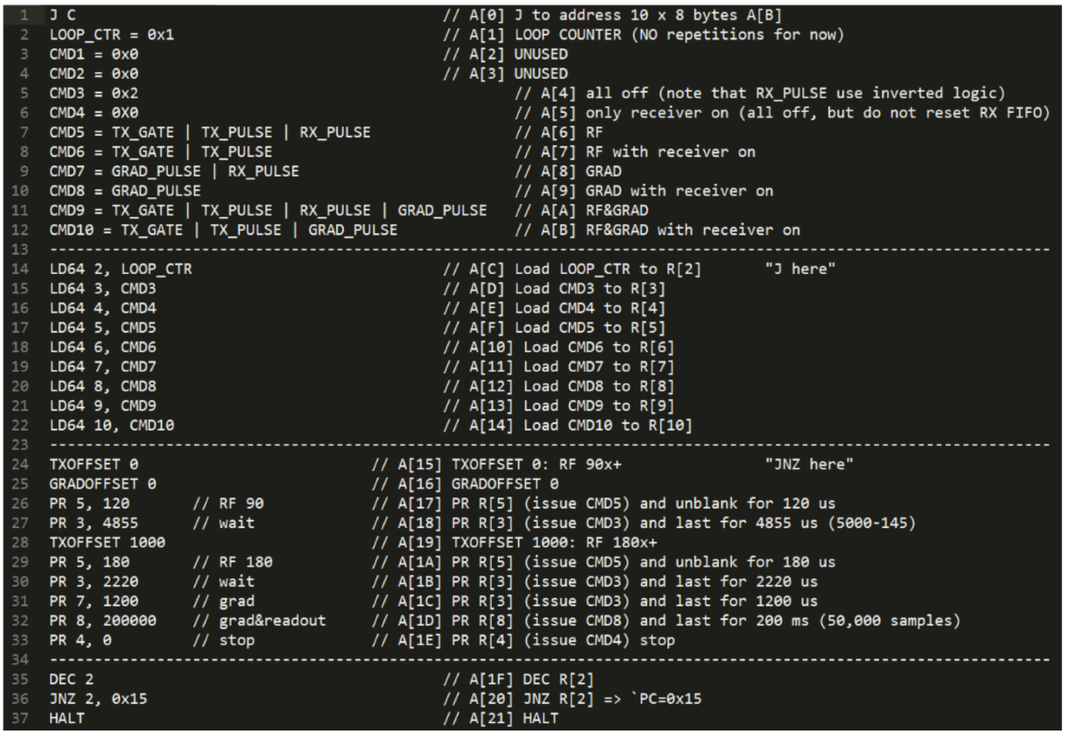
In the following part an usual implementation of a OCRA and Pulseq sequence is described.

3.2.1 OCRA

OCRA is a Python based interpreter for an assembly like language, specifically designed for the specification of timing in pulse sequence design for the RP [9]. The interpreter reads from a separated .txt file in which the timing of the pulse sequence is specified. For this purpose several commands are defined (tab. 3.4). To understand the functionality and implementation of the assembly like language an example is utilized of a 2D spin echo pulse sequence implemented in OCRA (fig. 3.7).

NOP	Does nothing
HALT	Ends the pulse sequence
DEC Rx	Decrement Rx
INC Rx	Increment Rx
LD64 Rx, addr	Load a 64 bit integer from addr to Rx
JNZ Rx, addr	Jump to addr if Rx!=0
J addr	Jump to addr
PR Rx, delay	Pulse Rx followed by a delay
TXOFFSET offset	Set offset of TX Pulse to offset
GRADOFFSET offset	Set offset of gradient pulse to offset
LITR delay	Inicate end of TR followed by a 40 bit delay
RASTCSYNC clkmask	Rest raster clocks indicated in clkmask

Table 3.4: All instructions defined by the assembly like language [18]



```
1  J C                                            // A[0] J to address 10 x 8 bytes A[B]
2  LOOP_CTR = 0x1                                 // A[1] LOOP COUNTER (NO repetitions for now)
3  CMD1 = 0x0                                     // A[2] UNUSED
4  CMD2 = 0x0                                     // A[3] UNUSED
5  CMD3 = 0x2                                              // A[4] all off (note that RX_PULSE use inverted logic)
6  CMD4 = 0X0                                              // A[5] only receiver on (all off, but do not reset RX FIFO)
7  CMD5 = TX_GATE | TX_PULSE | RX_PULSE                    // A[6] RF
8  CMD6 = TX_GATE | TX_PULSE                               // A[7] RF with receiver on
9  CMD7 = GRAD_PULSE | RX_PULSE                            // A[8] GRAD
10 CMD8 = GRAD_PULSE                                       // A[9] GRAD with receiver on
11 CMD9 = TX_GATE | TX_PULSE | RX_PULSE | GRAD_PULSE       // A[A] RF&GRAD
12 CMD10 = TX_GATE | TX_PULSE | GRAD_PULSE                 // A[B] RF&GRAD with receiver on
13 ------------------------------------------------------------------------------------------------------------------
14 LD64 2, LOOP_CTR                               // A[C] Load LOOP_CTR to R[2]       "J here"
15 LD64 3, CMD3                                   // A[D] Load CMD3 to R[3]
16 LD64 4, CMD4                                   // A[E] Load CMD4 to R[4]
17 LD64 5, CMD5                                   // A[F] Load CMD5 to R[5]
18 LD64 6, CMD6                                   // A[10] Load CMD6 to R[6]
19 LD64 7, CMD7                                   // A[11] Load CMD7 to R[7]
20 LD64 8, CMD8                                   // A[12] Load CMD8 to R[8]
21 LD64 9, CMD9                                   // A[13] Load CMD9 to R[9]
22 LD64 10, CMD10                                 // A[14] Load CMD10 to R[10]
23 ------------------------------------------------------------------------------------------------------------------
24 TXOFFSET 0                             // A[15] TXOFFSET 0: RF 90x+                 "JNZ here"
25 GRADOFFSET 0                           // A[16] GRADOFFSET 0
26 PR 5, 120       // RF 90               // A[17] PR R[5] (issue CMD5) and unblank for 120 us
27 PR 3, 4855      // wait                // A[18] PR R[3] (issue CMD3) and last for 4855 us (5000-145)
28 TXOFFSET 1000                          // A[19] TXOFFSET 1000: RF 180x+
29 PR 5, 180       // RF 180              // A[1A] PR R[5] (issue CMD5) and unblank for 180 us
30 PR 3, 2220      // wait                // A[1B] PR R[3] (issue CMD3) and last for 2220 us
31 PR 7, 1200      // grad                // A[1C] PR R[3] (issue CMD3) and last for 1200 us
32 PR 8, 200000    // grad&readout        // A[1D] PR R[8] (issue CMD8) and last for 200 ms (50,000 samples)
33 PR 4, 0         // stop                // A[1E] PR R[4] (issue CMD4) stop
34 ------------------------------------------------------------------------------------------------------------------
35 DEC 2                                          // A[1F] DEC R[2]
36 JNZ 2, 0x15                                    // A[20] JNZ R[2] => `PC=0x15
37 HALT                                           // A[21] HALT
```

Figure 3.7: Example of a 2D spin echo pulse sequence implemented in OCRA sequence[9, page 63]

.

The first 23 lines are equal in every case and are for the definitions of the commands. The registers contain afterwards the information about the kind of signal which is to be played out (RF, gradient). From line 24 onwards, the code defines the timing of the whole sequence, mainly by pulsing out the register, which defines the signal and adding some delays and timings, if needed.

Since the OCRA assembly like language only defines the timing of the sequence, the frequency and the shape of the signals are defined in Python code and get internally multiplied in the RPs FPGA [9].

The following code block shows an example of initialization of the experiment class (3.6) using the OCRA assembly like language.

```
from experiment import Experiment

exp = Experiment(
    samples=1256      #number of samples to
                          acquire during a shot
```

```
                         of the experiment
    lofreq=4.3576      #Larmor frequency
    tx_t=1             #RF TX sampling in microseconds
    rx_t=25            #RF RX sampling in microseconds
    instruction_file="/path/to/instruction/file.txt")

exp.add_tx(90_pulse)    #adds envelope for TX data
exp.add_tx(180_pulse)   #adds envelope for TX data
exp.add_grad(grad_x, grad_y, grad_z)     #adds gradient data

echo_samples = exp.run() # runs the pulse sequence
```

In the above example the instruction file contains the OCRA assembly like code and the methods exp.add_tx() and exp.add_grad() add envelope data to the corresponding outputs.

3.2.2 Pulseq

Pulseq describes the pulse sequences in a three level hierarchical structure which is stored in a .seq file.

The information of a hardware event is referenced via IDs in a row, which indicate to specific events in a different section, as exemplified by the following code.

```
<id><delay><rf><gx><gy><gz><adc><ext>
```

Every column specifies a different event. With the ID 0 indicating, that such an event is not contained in this part of the pulse sequence [19].

The <ext> entry may be used to define special use cases [19]

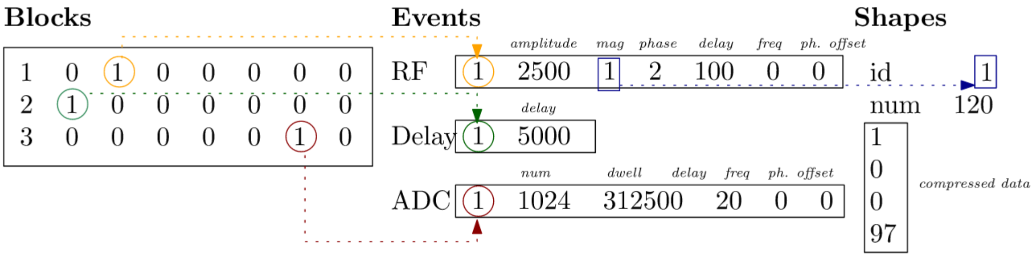


Figure 3.8: Visualisation of the hierachical structure in Pulseq[19, page 4]

In the event-sections, the event is defined in general parameters. Every column describing a new parameter.

If a shape definition is needed, an ID points to the shape description in yet another different section.

The shape itself is a list of samples in normalized range, describing the envelope of a signal in a compressed format. The compression procedure includes the calculation of the derivative of the shape and a subsequential compression for sections, in which the derivative stayed the same over more than two samples. In such a case, two samples are displayed followed by an integer, which indicates how many of that same number are following. [19].

Shape		Step 1 (derivative)		Step 2 (compression)
0.0		0.0		0
0.1		0.1		0.1
0.25		0.15		0.15
0.5		0.25		0.25
1.0		0.5		0.5
1.0		0.0		0.0
1.0	→	0.0	→	0.0
1.0		0.0		4
1.0		0.0		-0.25
1.0		0.0		-0.25
1.0		0.0		2
0.75		-0.25		
0.5		-0.25		
0.25		-0.25		
0.0		-0.25		

Figure 3.9: Example of the compression progress [19, page 13]

To translate Pulseq to the RP, an compiler called PSAssembler, is provided [16]. An example implementation in Python using Pulseq is demonstrated below.

```
from pulseq_assembler import PSAssembler
from experiment import Experiment

ps = PSAssembler(rf_center=lo_freq*1e6 #Larmor frequency)

tx_arr, grad_arr, command_bytes, output_dict
    = ps.assemble('path/to/instruction/file.seq')
```

```
exp = ex.Experiment(samples=params['readout_number'],
    lo_freq=lo_freq #Larmor frequency,
    tx_t=tx_t #RF TX sampling in micro seconds,
    rx_t=rx_t #RF RX sampling in microseconds,
    grad_channels=len(grad_arr))

exp.define_instructions(cb) #adds instructions
exp.add_tx(ps.tx_arr)       #adds envelope to TX data
exp.add_grad(ps.grad_arr)   #adds envelope to grad data

data = exp.run()      #runs the sequence
```

The above example shows, that the experiments class only takes the envelope from the .seq file and needs the information about the desired Larmor frequency. This might however change in the future, since the the PSAssembler class is still a prototype under development. The github repository can be visited with the following link.

`https://github.com/lcbMGH/ocra-pulseq`

In this fashion, Pulseq delivers a low level, but flexible language, which allows hardware independent pulse sequence developments [14]. The pulse sequence itself can be programmed in a high level programming language like matlab [14] or Python [15] or in a GUI, which lowers the threshold for beginners.

3.3 Experimental Methods

This section describes the measurement setup and and procedures that are used to evaluate the presented hardware modules of the signal chain

The complete signal chain that has been implemented and evaluated in this thesis is illustrated in figure 3.10. The hardware components used, operate within different frequency bandwidths. For the purpose of this work, the frequency 4.3576MHz is used, which corresponds to the Larmorfrequency of a low-field Halbach based permanent at $B_0 = 100$ mT magnet that was constructed in a separate work [8].

The figure 3.10 should also work as a construction guide. It lists all parts needed, which are described in the preceding sections and illustrates all the connections. The local machine refers to the laptop or desktop PC which is used as interface to control the RP.

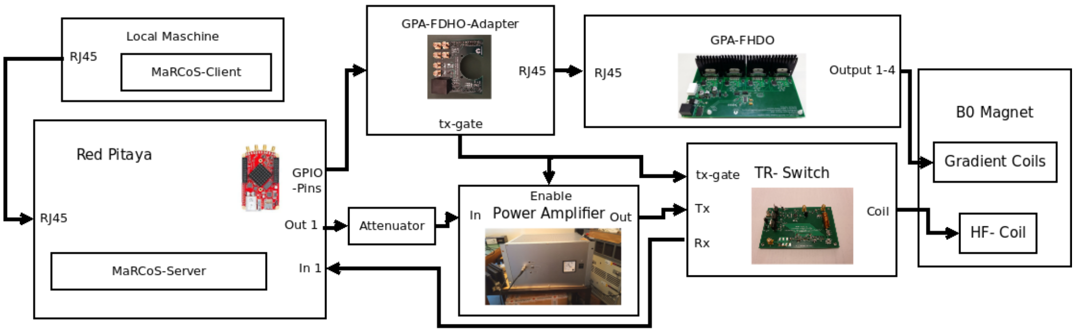


Figure 3.10: Schematic of the hardware components and interfaces of the signal chain evaluated in this work

$$f_{\text{Larmor}} = 4.3576\text{MHz} \tag{3.1}$$

3.3.1 General approach

To quantify the performance of the signal chain modules, some important metrics are introduced.

- Transmit pulse frequency fidelity
- Fidelity of the RF and gradient pulse shape
- Flip angle uncertainty
- Timings between RF and gradient pulses

These parameters are important factors that influence the final imaging results. The evaluation is performed assuming a standard spin echo sequence with rectangular $\frac{\pi}{2}$ and π RF pulse and corresponding gradient pulses.

An RTO 1022 oscilloscope (Rhode and Schwarz) is used for all the measurements.

The figure 3.11 illustrates the signal flow in the MRI scanner. The results contain measurements starting from the RP. The measurements are therefore generally separated into the RF signal chain and the gradient signal chain.

To identify potential deviations from output vs input signals of the hardware modules in the signal chain, the signal is measured before and after every module.

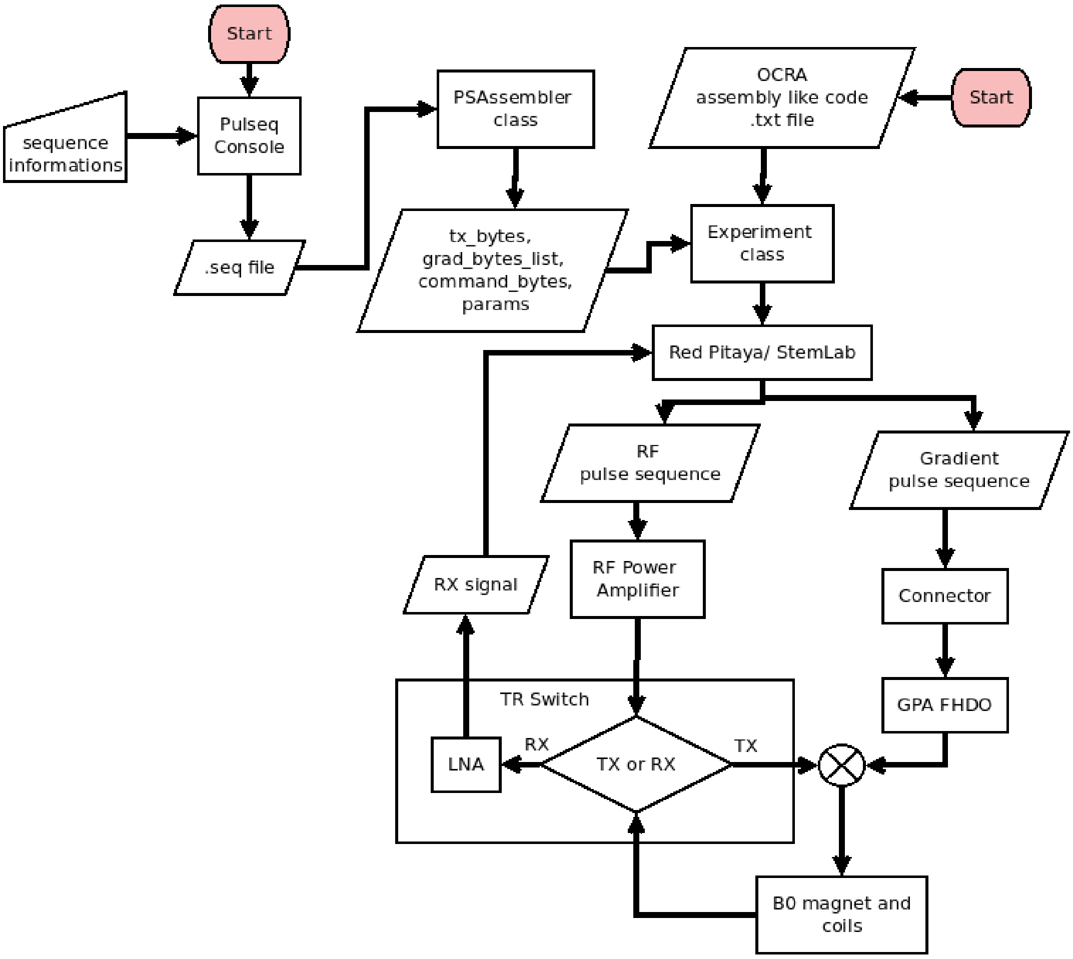


Figure 3.11: Signal flow of the measurement setup

The Pulseq console in figure 3.11, used in this work is the Matlab programm called tabletop_se_1d_pulseq.m from the MaRCoS_experiments repository in table 3.7.

To ensure a correct measurement, 10 measurements are taken, and evaluated to a mean and uncertainty value.

Variables and plots

To evaluate the robustness of the system, T_E and the duration of the $\frac{\pi}{2}$ pulse (rf90duration) are varied to create a subsequently compact pulse sequence (table 3.5).

parameter	values
T_E	[6, 8, 10, 12]ms
rf90duration	[10, 20, 40, 60, 80]μs

Table 3.5: Measured values

A further decrease of T_E is prevented by an exception in Matlab of tabletop_se_1d_pulseq.m, since a required delay, between the gradient pulses could not be met.

The shape of the RF as well as the gradient pulses are chosen to be rectangular. This has the advantage that the amplitude and width of the individual pulses define the shape of the signal. For this reason the amplitude of the signals are also varied. Additionally to this, the measurements are examined if the shapes contained any irregularities, which deviate from the rectangular shape.

In the measurements, the outputs of the individual components are put in relation to the values set according to equation (3.2). The mean value of all measurements is printed as a red horizontal line.

In the voltage amplitude measurements, the transfer behavior of the signal chain, up to the module which is measured, is subsequently analysed, to predict the output amplitude of the evaluated module. This prediction of the amplitude is called V_{set} and deviations from it are evaluated.

$$\begin{aligned} \Delta f &= f_i - f_{\text{Larmor}} \\ \Delta V &= (V_i - V_{\text{set}})/V_{\text{set}} \cdot 100 \\ \Delta t &= t_i - rf90duration \\ f_{\text{Larmor}} &= 4.3576MHz \end{aligned} \tag{3.2}$$

Standard values

If not otherwise stated, standard values for T_E, rf90duration and the amplitude of the RF pulse (V_{RF}) are used to ensure a comparability of the results (tab. 3.6).

parameter	**value**
V_{RF}	312.5 mV
rf90duration	80μs
T_E	12 ms

Table 3.6: Standard values during the measuerments.

Repositories

The measurements, except the OCRA based RP output, are all based on a Pulseq encoded 1d spin echo pulse sequence implemented in a program which is provided by Benjamin Menküc. This program can be found in his MaRCoS Experiments repostitory. For general installation instructions refer to the github wiki of Vlad Negnevitskys repository, MaRCoS Extras [20]. The table 3.7 lists the repositories used for the measurements.

Repository	**Branch**	**Commit**
MaRCoS Extras	scratch	95bc65c9cf2b3ba107df11cb1ce0a55f1c9699eb
MaRCoS Client	ocra_grad_ctrl	3a1ea9bddb6b96139cbff144b53d16bc943e6d93
MaRCoS Experiments	master	137b1e15b261d6f476a7c39c42bf19ec9f4358ad
OCRA	new_multiplier	88f468ada059f69ccb3cec28c638fa929822a889
OCRA Pulseq	grad-dev	5682400aa1f427abacc46f5ac8a961d6ab87c6d3

Table 3.7: Respositories and Branches for recreation of the measurements.

All repositories listed in the table 3.7 can be found under the following link.

`https://github.com/catkira?tab=repositories`

The OCRA encoded pulse sequence used in the measurements is provided by Ruben Pellicer. The source code can be found under the following link.

`https://github.com/RPellicer/marcos_experiments`

3.3.2 Data analysis

Automated scripts are implemented in Python to analyze the data. The dominant frequency, amplitude and width of the signals, as well as the time between the fist and second pulse are automatically extracted from the data. Since the signal pulses are chosen to be rectangular, the width and amplitude of the signal define the shape of the signal.

To increase the sampling rate in the frequency domain to $\frac{1}{20\mathrm{Hz}}$, the measured signals are extended with zeros and then transformed with a FFT. Subsequentially the highest amplitude between 3 MHz and 5 MHz is evaluated and the according frequency stored.

Before the amplitude and width of the signals are evaluated the signals are high-pass-filtered to dampen noise.

In order to calculate the amplitude, the first derivative of the measured signal is calculated. If the signal is positive, over 80 % of the maximum amplitude, and its derivative is under 1% of its maximum, the amplitude is stored. If more than 5 samples are gathered, a mean value is generated and stored. The threshold serves as an additional noise filter.

To calculate the width of the signal, an envelope of the signal is generated. Two time stamps are recorded. The fist is when the amplitude reaches 90% of its set value. And the other is recorded when the signal falls to back to 10 % of its set amplitude. The difference is then stored. This procedure is essential in the evaluation of the RF pulse width of the RFPA and TR-switch, since the RFPA generates a short pulse when triggered by the gate pulse.

To calculate the time period between the second and first pulse, the time in which the amplitude of the measured signal rises over 60 % of its maximum amplitude, in each pulse is recorded and then a difference is made.

All evaluated information are stored in .txt file in the same folder as the measurements.

4 Results

In the following the functionality of the MRI scanner signal chain is analysed. All measurements as well as the evaluation software can be downloaded under the following link.

`https://github.com/ooggesen/BA`

4.1 Results in RF signal chain

In this section the fidelity is tested before and after every module in the signal chain (figure 3.11).

4.1.1 Red Pitaya

OCRA based RF pulse generation

The following measurement, visualized in figure 4.1, 4.2 and 4.3, shows an OCRA encoded pulse sequence, played out by channel 1 of the RP. All measured and set parameters are defined in table 4.1, with the rf180duration standing for the duration of the RF π pulse, which was accidentally set to 200μs in the assembly like OCRA code.

A comparison of the modelled pulse shape and the measured pulse shape after the RP output is visualizes by a ideal pulse shape. The measured pulse shape shows deviation in the amplitude compared to the theoretical one.

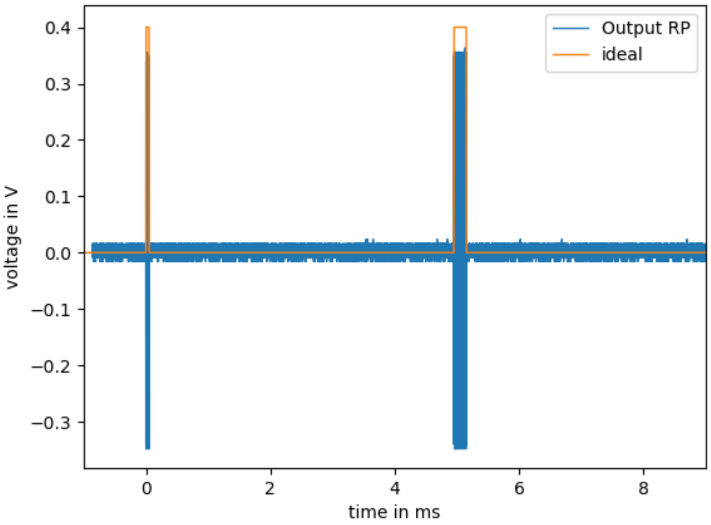


Figure 4.1: Measured $\frac{\pi}{2}$ and π pulse generated by OCRA and the RP

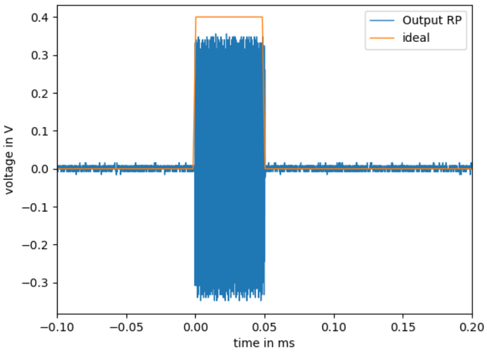


Figure 4.2: Zoom over figure 4.1

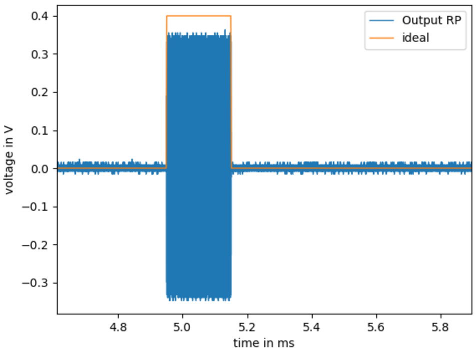


Figure 4.3: Zoom over figure 4.1

parameter, set	**value**
$V_{out,RP}$	400 mV
$rf90duration$	50μs
$rf180duration$	200μs
T_E	10 ms
parameter, measured	**value**
$V_{out,RP}$	343.1 ± 0.03mV
f_i	4357596 ± 4.67Hz
$rf90duration$	$50.92 \pm 0.04\mu$s

Table 4.1: Measured and set parameters for OCRA based measurement.

Pulseq based RF pulse generation

The following figures are visualizations of a Pulseq encoded pulse sequence of measurements generated by the RP. The figure 4.4, 4.5 and 4.6 shows an example $\frac{\pi}{2}$ and π pulse and an ideal shape, for comparison. The measured pulse shape shows good fidelity compared to the theoretical one. All set parameters are listed in table 4.2.

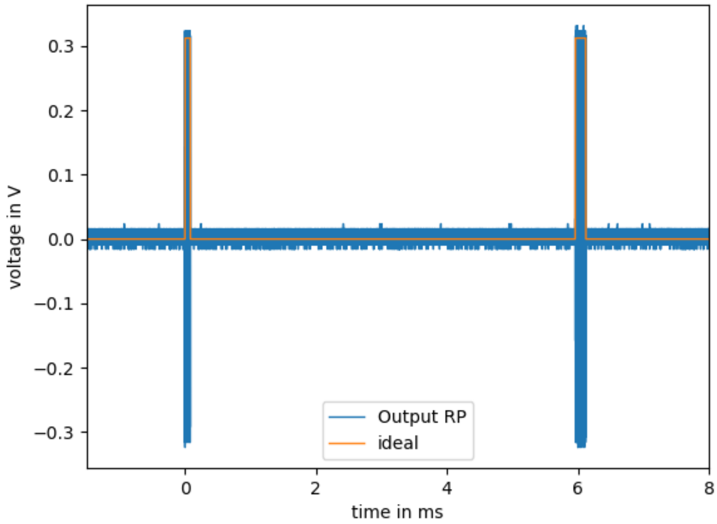


Figure 4.4: Measured 90 and π pulse generated by Pulseq at the output of the RP

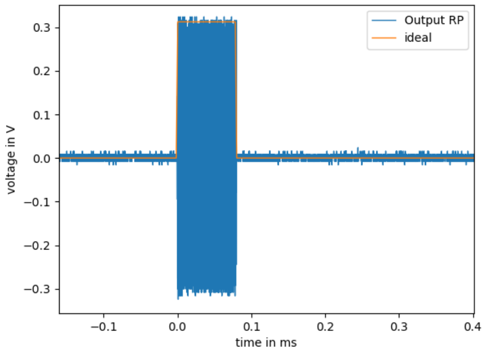


Figure 4.5: Zoom over figure 4.4.

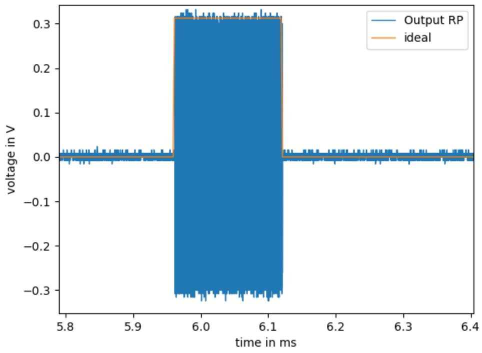


Figure 4.6: Zoom over figure 4.4.

$rf90duration$	80μs
V_{set}	312.5 mV
T_E	12 ms

Table 4.2: Parameters set for Pulseq encoded pulse sequence

The figure 4.7 evaluates the frequency over the width of the $\frac{\pi}{2}$ pulse, to see if there are any switching speed limits to the RP. The measurements have a maximum frequency deviation at a pulse width of 10 μs of -0.83 kHz, but with longer duration of the $\frac{\pi}{2}$ pulse, more samples are available to the FFT and the deviations decrease to 110 Hz, above the set frequency. If the measurement at 10 μs pulse width is excluded, the frequency deviations results in a mean deviation of 187.5 Hz with a standard deviation of 17.7 Hz.

The The frequency depicted in figure 4.8 in contrast, takes frequency information of the 90 and π pulse and evaluates it over T_E, to visualize switching speed limits, if two RF pulses follow each other in a fast pace. The measurements, with mean deviation of 0.067 kHz and according standard deviation of 0.062 kHz, are similar to the measurement in figure 4.7 at RF pulse duration of 80 μs, which indicates, that the RF pulses are not distorted by fast succession.

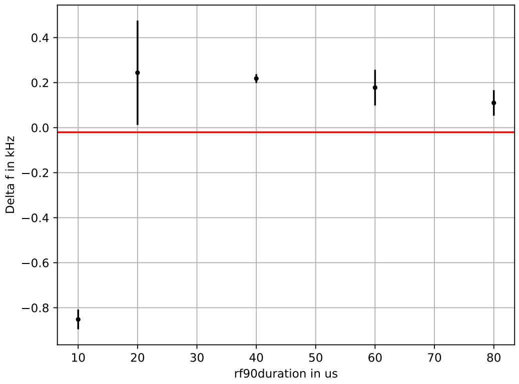


Figure 4.7: Frequency difference over rf90duration of $\frac{\pi}{2}$ pulse, output from the RP

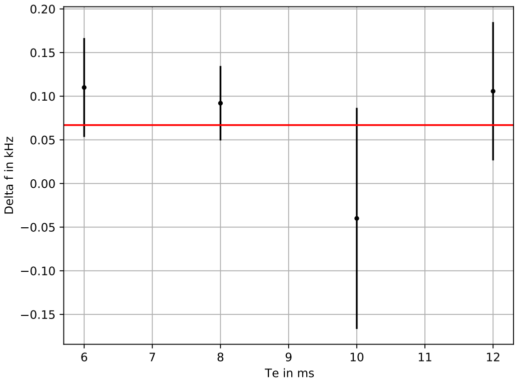


Figure 4.8: Frequency difference over T_E of the 90 and π pulse, output of the RP

The figures 4.9 and 4.10, analyse the shape of the $\frac{\pi}{2}$ pulse generated by the RP. At amplitudes greater than 400 mV an overflow error in Python occured. The result is the fall of the amplitude in figure 4.9 at amplitudes over 400 mV. Up to 400 mV amplitude, the measured deviations have a mean deviation of 2.68 % with a standard deviation of 0.086 %, which shows the high amplitude fidelity of the RP.

The figure 4.10, visualizes deviations in the pulse width of the $\frac{\pi}{2}$ pulse, with a mean deviation of -0.82 μs and according standard deviation of 0.016 μs. Together with the measurements from the figure 4.9, the RF output of the RP shows high fidelity over amplitudes of $0.05 - 0.4$V and over widths of $10 - 80\mu$s.

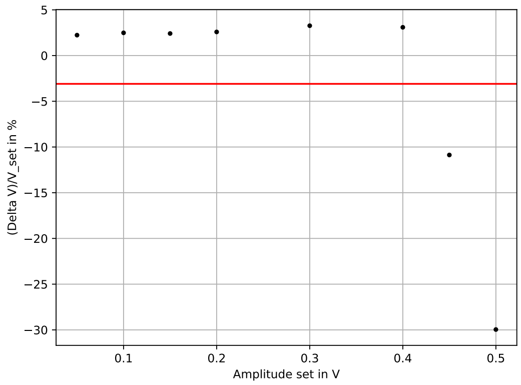


Figure 4.9: Amplitude difference over amplitude set of $\frac{\pi}{2}$ pulse, output of the RP

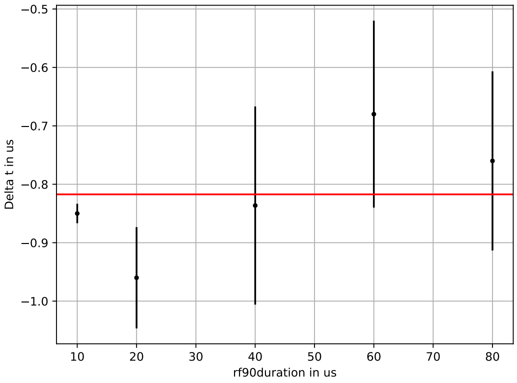


Figure 4.10: Deviations of the measured pulse width of the $\frac{\pi}{2}$ pulse over RF pulse width set, output of the RP

4.1.2 RF power amplifier

The figure 4.11 visualizes the output of the RFPA, with the gate pulse. A zoom into the spike pulses at the start and end of the gate pulse is depicted in the figures 4.12 and 4.13.

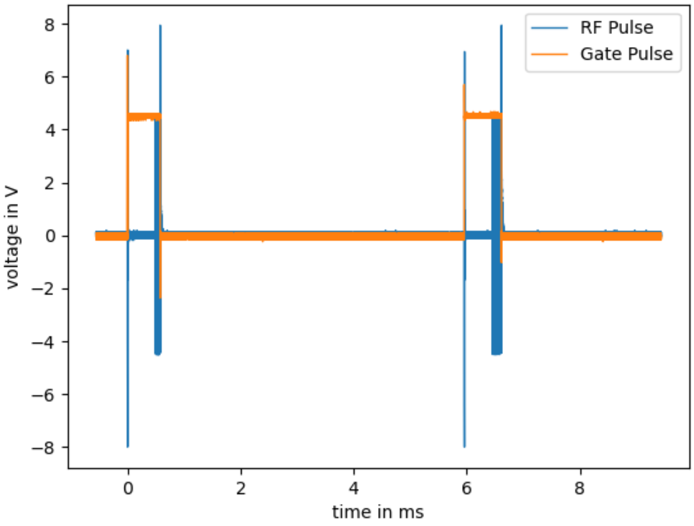


Figure 4.11: 90 and π pulse at the output of the RFPA. Amplitude set 4.4 V

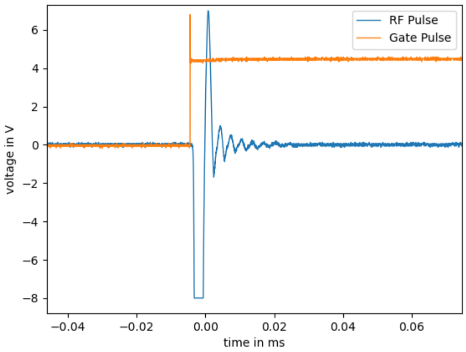


Figure 4.12: Settling behavior RFPA at start of gating pulse.

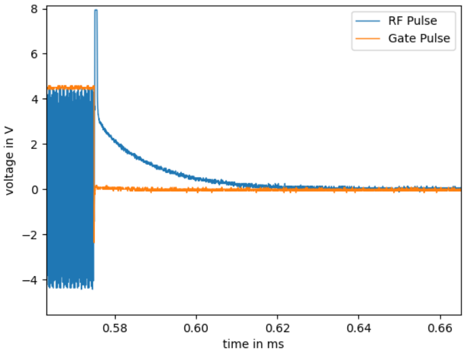


Figure 4.13: Settling behaivior at end of gating pulse.

The figure 4.14 visualizes the measured transfer behavior of the RFPA in comparisson with an ideal set amplification of A_{RFPA}= 500. The measurements follow this amplification, with mean deviations of 47 mV. Therefore the set amplification of 500 is used to calculate the output amplitude set in the figure 4.17.

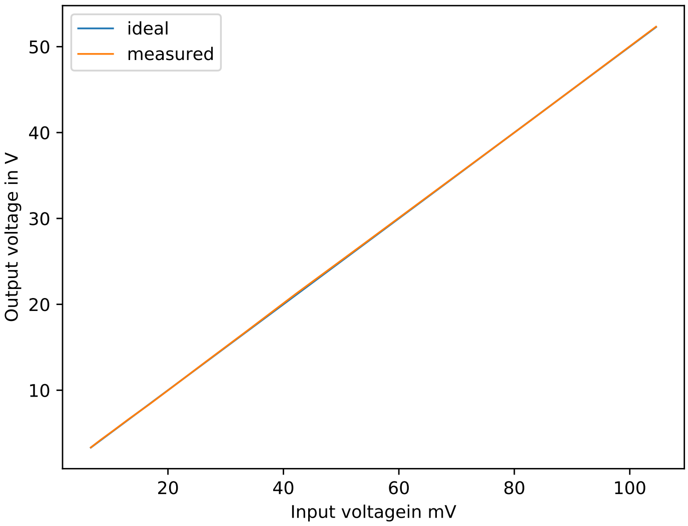


Figure 4.14: Transfer behavior of the RFPA.

The figures 4.15 and 4.16 are constructed in the same manner as the figures 4.7 and 4.8, to evaluate if the RFPA changes the frequency of the pulses generated by the

RP. The figure 4.15 shows a similar trajectory as the 4.7 and if the measurement with the duration of the $\frac{\pi}{2}$ pulse of 10 μs is excluded, the measurements have a mean deviation of 39.5 Hz with a standard deviation of 90 Hz. The figure 4.16 displays measurements with a mean deviation of 92.5 Hz and according standard deviation of 18.8 Hz. Both are inline with the measurements of the RP, which indicates, that the the RFPA has high frequency fidelity.

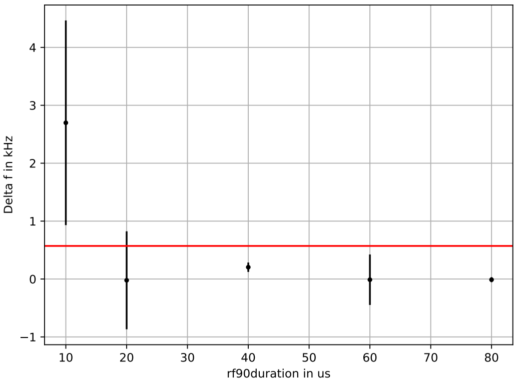


Figure 4.15: Frequency difference over the duration of the $\frac{\pi}{2}$ pulse.

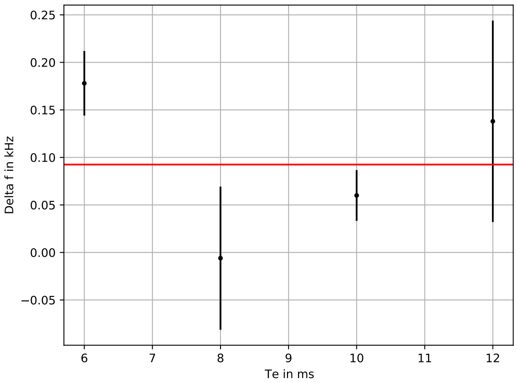


Figure 4.16: Frequency difference over T_E

The amplitudes in the figure 4.17 were varied with attenuators put before the RFPA (fig. 3.10). This procedure is taken to prevent the overflow error from the RP at amplitudes over 400 mV depicted in figure 4.9. With a mean of 0.23 % and standard deviation of 0.21%, the gain of 500 of figure 4.14 and the fidelity of the RFPA is appoved.

The figure 4.18 examines deviations in the measured width of the $\frac{\pi}{2}$ pulse, over the pulse width set, similar to the figure 4.10. With a mean of -1.53 μs and a standard deviation of 0.06 μs, the Pulse width of the RFPA seem to have decreased relative to the RP for about 0.7μs, which produces no significant change in the the flip angle, that it can be ignored.

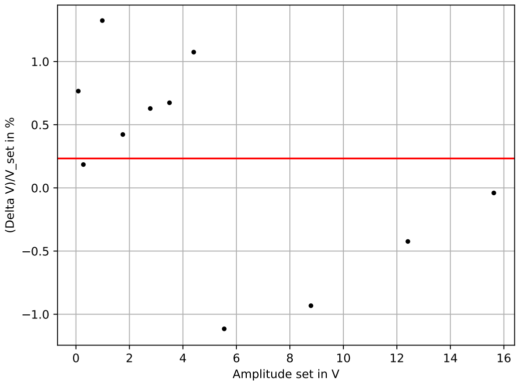


Figure 4.17: Amplitude difference over amplitude set of the $\frac{\pi}{2}$ pulse, output by the RFPA

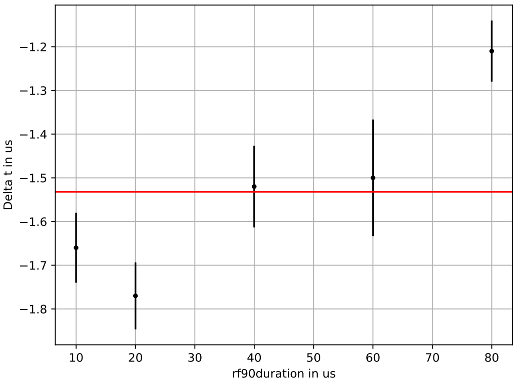


Figure 4.18: Pulse width of the $\frac{\pi}{2}$ pulse over RF pulse width set, output of the RFPA

4.1.3 TR-switch

In the following measurements the results are separated into the TX and RX use case.

TX to coil

The figures 4.19, 4.20 and 4.21, show a sample output of the TR-switch at the coil adapter.

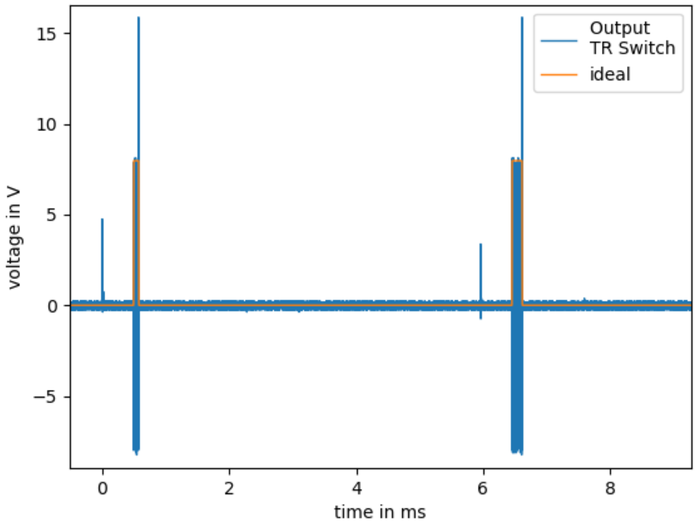


Figure 4.19: $\frac{\pi}{2}$ and π pulse at the Coil adapter of the TR-switch. Amplitude set 7.9 V

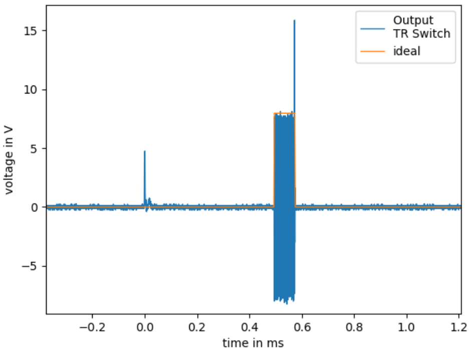


Figure 4.20: Zoom over $\frac{\pi}{2}$ pulse in figure 4.19

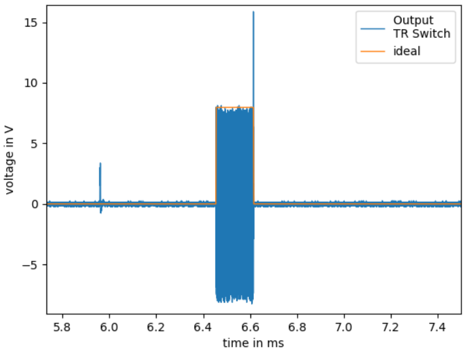


Figure 4.21: Zoom over π pulse in figure 4.19

To predict the behavior of the TR-switch, the scattering parameters at the system frequency of 4.3576 MHz were measured, which are displayed in table 4.3. The TX to coil gain (5V) is used to calculate the voltage set at the Coil, illustrated in figure 4.22.

Parameters (tx_gate)	**measured values**
TX to RX gain (0V)	−17.9dB
coil to RX gain (0V)	38.43dB
TX to RX gain (5V)	+0.9dB
TX to coil gain (5V)	−0.85dB
TX to RX isolation	−37.53dB
PI network	
capacity	740pF
inductance	2.7μH

Table 4.3: Parameters measured of the TR-switch used

The TR-switch has protective diodes, which switch at a voltage level of 7 dBm at the second LNA input, build into the RX signal path, which causes a increased deviation at amplitudes over 8 V at the figure 4.22. If the two measurements at set amplitudes over 8 V are excluded, the TR-switch shows deviations with a mean of 4.2 % and standard deviation of 0.08%, which indicates, that the TX to Coil gain (5V) from the table 4.3 might be lower.

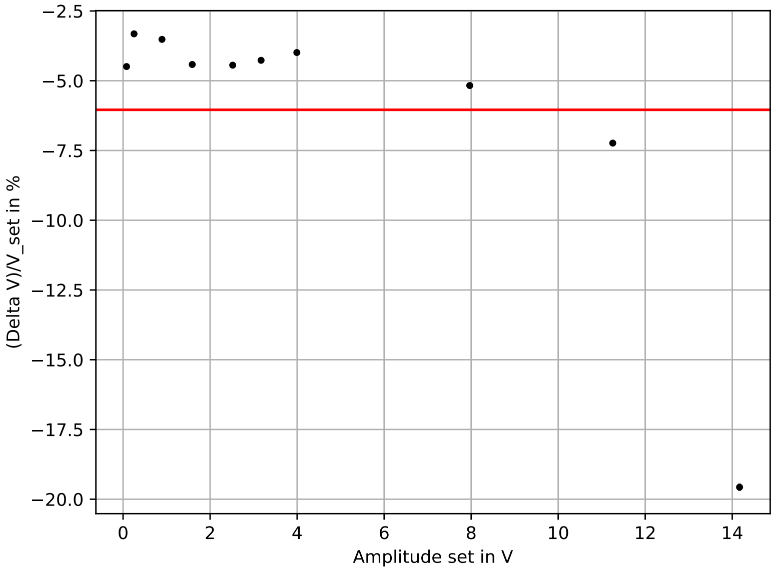


Figure 4.22: Amplitude difference over amplitude set.

Coil to RX

To evaluate the gain of the RX signal path, the pulse sequence generated by the RP, specified by the table 4.2, is attenuated and that fed into the TR-switch. The figure 4.23 displays a sample output of the RX path, with a $\frac{\pi}{2}$ and π RF pulse.

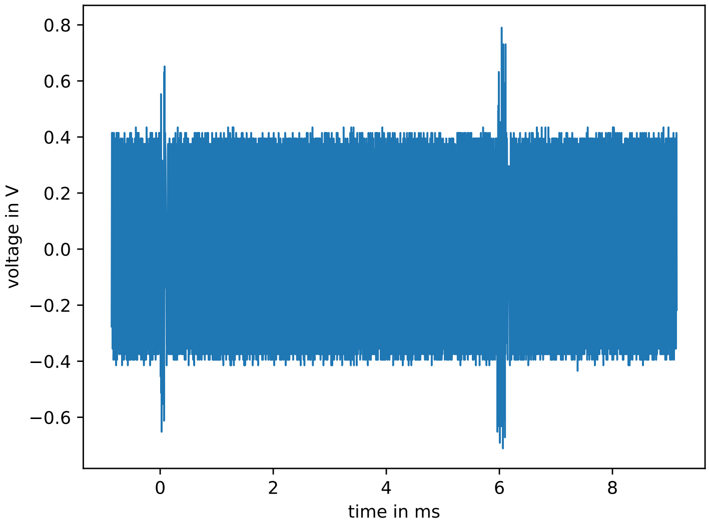


Figure 4.23: Voltage at RX port with 2.47 mV at coil port.

The figure 4.24 shows a FFT of the amplified noise of the RX path, with figure 4.25 depicting a zoom over the system frequency. The measurements were taken, without a low-pass-filter, such that the results might include artifacts of aliasing, due to insufficient sampling frequency.

The figure 4.25 shows a frequency peak at ≈ 2.6MHz, which might alter signals, if the scanner frequency is chosen to be in that frequency bandwidth.

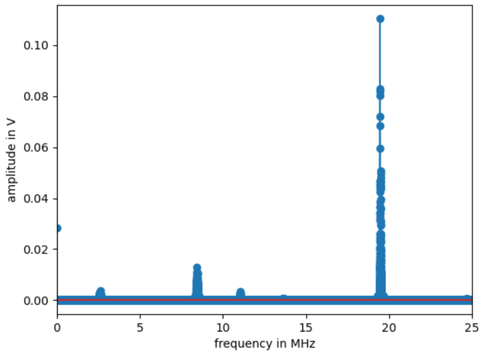


Figure 4.24: FFT of measured noise at output of RX signal chain of the TR-switch

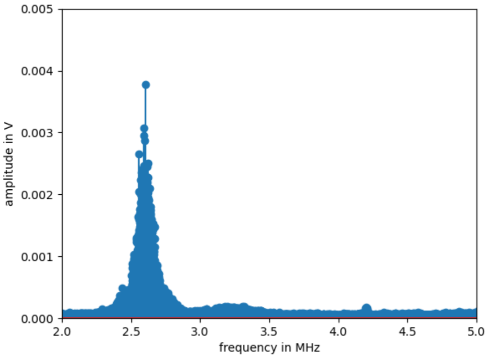


Figure 4.25: FFT of measured noise at output of the RX signal chain of the TR-switch, zoomed in

The figure 4.26 visualizes the measured amplification of the RX signal path, over different input amplitudes. Lower amplitudes could not be measured, since the pulses, had a lower amplitude than the noise depicted in figure 4.19. With a mean amplification of 44.2 dB and a visual increase in amplification at lower input amplitudes, the amplification of the RX signal chain might be even higher at the voltage level of a SE, which is usually in scale of a μV

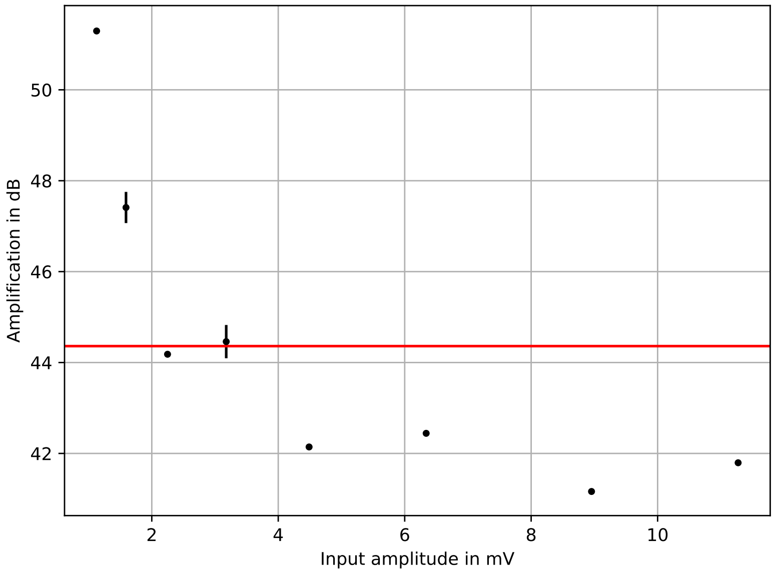


Figure 4.26: Measured amplification of LNA in the RX path of the TR-switch.

4.2 Results in gradient signal chain

The following measurements evaluate the behaivior of the GPA-FHDO and the timing of the RF signal chain and gradient signal chain outputs.

4.2.1 Linearity

In the Pulseq program, only 3 different gradients could be programmed. Therefore only the first three outputs were analysed. But since all outputs are similar, the behavior should be the same for the fourth available output.

To calculate the current amplitudes of the gradient amplifier the voltage is measured over a 0.1 Ω resistor and translated to current information with Ohms law.

The figure 4.27 visualizes a sample output of the GPA-FHDO together with an ideal shape. This shows high amplitude fidelity of the GPA-FHDO, as well as sufficient rise times.

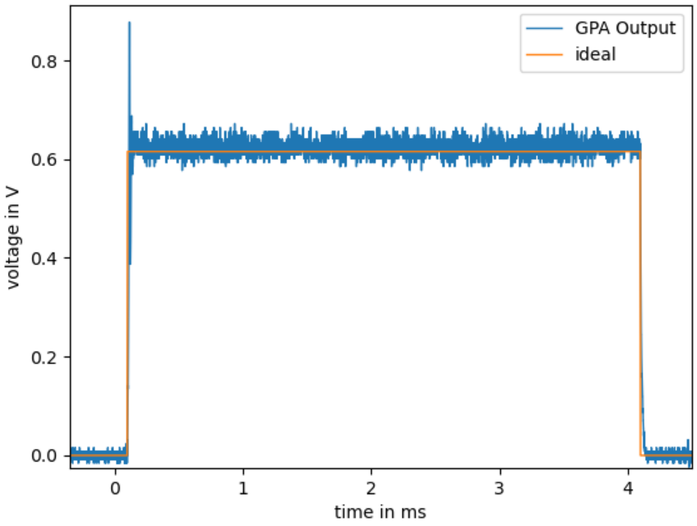


Figure 4.27: Gradient pulse with Amplitude of 6.1 A and width 4 ms. Measured over a 0.1Ω resistor.

The figures 4.28, 4.29 and 4.30 show deviations of the measured amplitude with respect to the amplitude set of the Channels 1 to 3. With mean deviations under 2 % the GPA-FHDO shows high fidelity in all Channels.

The figures 4.31, 4.32 and 4.33 show deviations of the gradient pulse width over the amplitude set. In this measurements the pulse width of the gradient pulses were set to 4 ms and the measurements aim to reveal shape distortions. With mean deviations over -9.5 μs the GPA-FHDO shows high fidelity even at low current amplitudes.

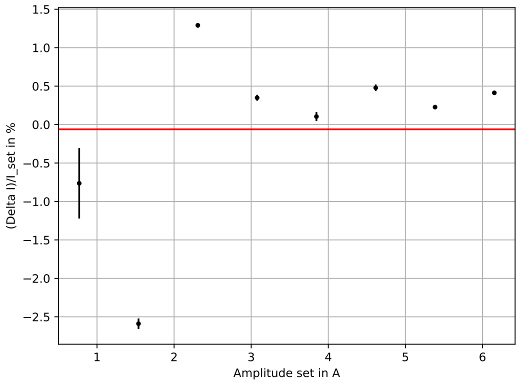


Figure 4.28: Amplitude difference over amplitude set at Channel 1 of GPA-FHDO

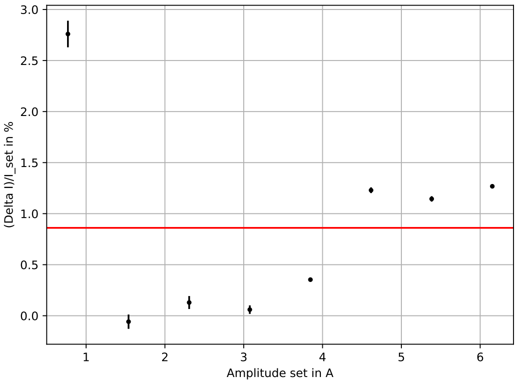


Figure 4.29: Amplitude difference over amplitude set at Channel 2 of GPA-FHDO

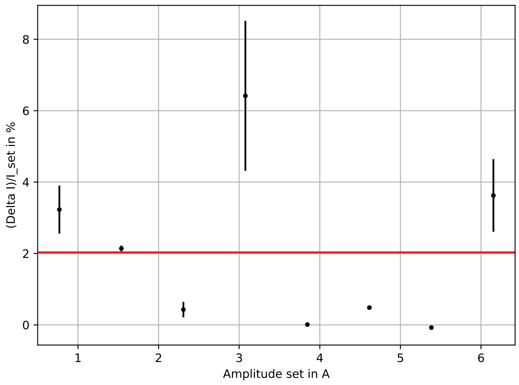


Figure 4.30: Amplitude difference over amplitude set at Channel 3 of GPA-FHDO

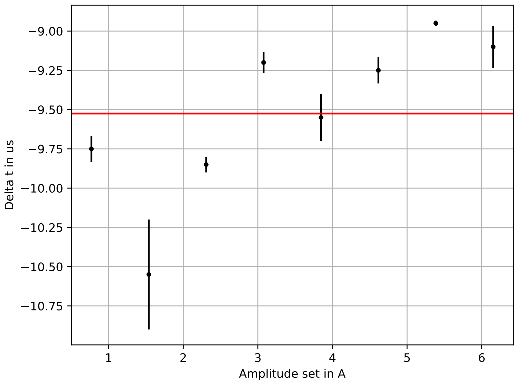


Figure 4.31: Width difference of gradient pulses over amplitude set at Channel 1

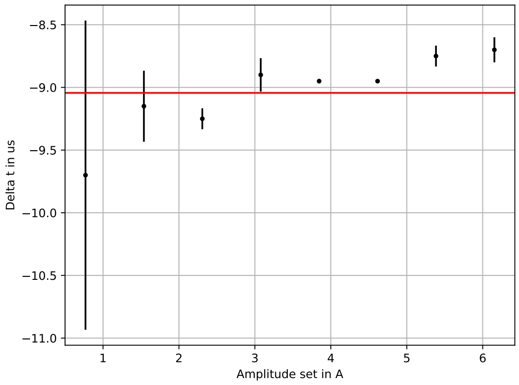


Figure 4.32: Width difference of gradient pulses over amplitude set at Channel 2

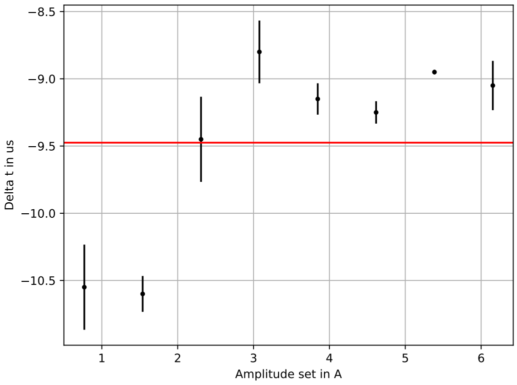


Figure 4.33: Width difference of gradient pulses over amplitude set at Channel 3

4.2.2 Timing

The figures 4.34 and 4.35 show sample scanner outputs, with T_E set to 6ms or 12ms, containing a $\frac{\pi}{2}$ and π RF pulse and a dephasation and frequency encoding gradient pulse. In both cases the timing of the gradients and RF pulses align with the set T_E. The same measurements were repeated for T_E set to 8 ms and 10 ms, with similar results. In all measurements the timing of the scanner does not seem to be impaired.

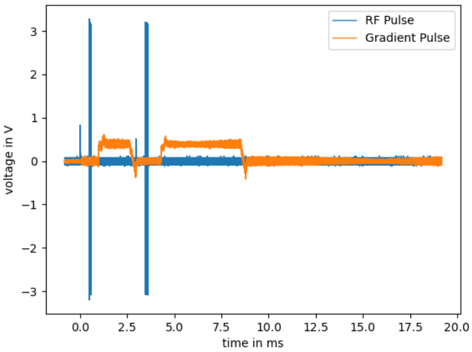


Figure 4.34: Full scanner ouput. $T_E = 6$ms

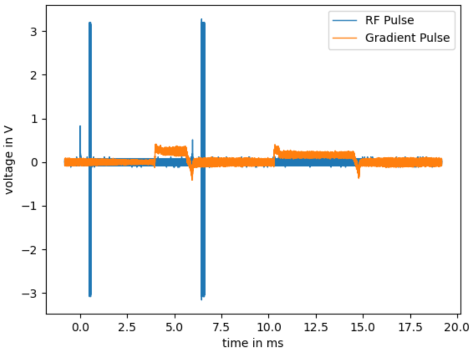


Figure 4.35: Full scanner output. $T_E = 12$ms

5 Discussion

In this section the results are discussed and alternative hardware modules are presented.

5.1 RF signal chain

This sections evaluates the results of the RF signal chain., differentiated into the individual modules in the scanner setup.

5.1.1 Red Pitaya

The OCRA/MaRCoS system, running on the RP currently supports the OCRA and Pulseq pulse sequence defining language. But since Pulseq establishes itself as a cross vendor standard, its development and support greatly exceeds that of OCRA. Following this reasoning, this work focused on the Pulseq defined pulse sequence. The measurements of the OCRA encoded pulse sequence therefore serves as an example of compatibility. The performance however is therefore evaluated with the Pulseq encoded pulse sequence.

The results show a deviation of the RF output of the RP at high amplitudes. This occurs with the Pulseq encoded pulse sequence, with amplitudes over 400 mV, but the OCRA encoded pulse sequence, shows similar behavior at the output amplitude of 400 mV. This suggests, that the full RF output swing of the RP is not exploitable.

Other than this occurrence, the RP provides high accuracy in the frequency domain and shape of the RF signals, even with very short RF pulses as shown in the figures 4.7, 4.8 and 4.10 .

In comparison to the URSP X300 [21] which is capable of playing out RF pulses with up to 6 GHz frequency, the RP provides sufficient functionality but comes at a much lower price with ≈ 500 € compared to ≈ 5000 € of the URSP X300, suggesting the usage of the RP in a MRI scanner implementation.

If however the capability of the URSP X300 series, or other SDRs with different functionality, is needed, the RP could be replaced with a software package called grMRI, which is utilizing the GNU radio toolkit [22]. This however might result in higher costs and support for the other components in the measurement setup (figure 3.10) are not provided.

5.1.2 RF power amplifier

With the voltage spikes at the output of the RFPA, having a frequency of ≈ 150 kHz, the RFPA performed equally well in the frequency domain about the system frequency as the RP. This suggests that the frequency offsets illustrated in figure 4.7 and 4.8 are caused by the RP. Together with the measurements from figure 4.14 and 4.17 the results suggests high fidelity of the RFPA.

However, a limit to the RFPA output was set to 16 V, to prevent damage to the TR-switch which precedes the RFPA. This resulted in a maximum output power over a 50 Ω resistor of 2.56 W. With a maximum load power of 1 kW, the amplification of the RFPA in the measurements is by orders of magnitude higher than what would be suffice for this scanner setup. This suggests, that the RFPA should be replaced by a cheaper and more compact, but less powerful solution. Since open source RFPA designs are scarce, a proprietary design might be feasible.

5.1.3 TR-switch

The TR-switch, which design is based on the TR-switch published in the tabletop MRI system at MIT [23], contains two separated signal chains. First in order, the TX to coil signal chain is analysed.

The measured stray parameters presented at the table 4.3 and the measurements from figure 4.22 indicate, that the TR-switch, can conduct up to 8 V input amplitude over the TX signal path. This could even improve, if the inductance of the coil in the pi network of the TR-switch is more precisely tuned. This sets limits to performance of the system, but with the RF signal having a maximum power of 640 mW over a 50 Ω resistor, it should suffice for low field MRI systems.

Second in order is the TR-switch coil to RX signal chain, which is used in the receive case and which amplifies the RF signal received at the Coil port. The amplification of this path could not be measured for amplitudes in the μV scale due to high noise. But the figure 4.26 suggests that the amplification exceeds 44dB. The noise could also be filtered since it has its closest peak to the system frequency lies at $\approx$ 2.5MHz (figure 4.24), which is sufficiently appart from the Larmor frequency of the system. With the RP detecting signals up to U_{LSB} of $\approx 7.6\mu$V input amplitude, the amplification should suffice.

The noise however provides problems, provided it is not caused by Aliasing, if the desired Larmor frequency is in the same spectrum. This makes the TR-switch unusable for B_0 field strength of $\approx$ 69mT.

5.2 Gradient signal chain

During the time when this thesis was written, the software of the MaRCoS system went under major changes, which were not backwards compatible. This resulted in minor incompatibilities between the software repositories in the measurements and in turn to deviations in the current amplitude set. This was compensated for by using a current transfer function, which was evaluated by measurements.

And with current deviations in mean under 2.03 % (mean value from fig. 4.30), the GPA FHDO provides high linearity in all 3 Channels. The figures 4.31, 4.31 and 4.33 also shows high accuracy in the shape of the signal, as displayed by an example in figure 4.27.

If more than 4 current outputs are necessary, the current driver by Nick Arango et al. [24] might be a better fit. With a lesser current driving capabilities of 8 A compared to the GPA FHDO, it provides a scalable solution at only 75 US$ per channel.

The speed of the GPA is evaluated in the chapter 4.2.2. As the figure 4.34 and 4.35 shows, the shape of the pulses is not disturbed by a fast succession of gradient pulses. This holds true for all ten measurements of the most compact pulse sequence, with T_E= 6 ms.

5.3 Conclusion

The performance and timing of the scanner setup fulfills the set requirements of a low field MRI scanner. In conclusion this suggests, that the RP is capable of controlling the peripheral modules to generate synchronized RF and gradient pulses of sufficient amplitudes and of a desirable shape. With this implementation the utilisation for diagnostic usage in MR imaging of extremities seems possible. This should suffice as proof of concept, that a MRI scanner can be implemented on the bases of open source soft- and hardware, resulting in the development of MRI scanners with broader functionalities.

Bibliography

1. Brown, R. W., Cheng, Y. N., Haacke, E. M., Thompson, M. R. & Venkatesan, R. *Magnetic Resonance Imaging* 1–17. ISBN: 978-1-118-63395-3 (John Wiley & Sons, Ltd, 2014).

2. Marques, J. P., Simonis, F. F. J. & Webb, A. G. Low-Field MRI: An MR Physics Perspective. *Journal of Magnetic Resonance Imaging* **49,** 1528–1542 (2019).

3. Wald, L. L., McDaniel, P. C., Witzel, T., Stockmann, J. P. & Cooley, C. Z. Low-Cost and Portable MRI. *Journal of Magnetic Resonance Imaging* **52,** 686–696 (2020).

4. Pearce, J. M. Cut Costs with Open-Source Hardware. *Nature* **505,** 618–618 (**january** 2014).

5. Caverly, R. H. *PIN Diode-Based Transmit-Receive Switch for 7 T MRI* **in**. 2016 IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireleSS) (**january** 2016), 100–102.

6. Brunner, D. O. **andothers**. Symmetrically Biased T/R Switches for NMR and MRI with Microsecond Dead Time. *Journal of Magnetic Resonance* **263,** 147–155 (**february** 2020).

7. *Introduction to NMR/MRI RF Amplifiers* **may** 2016. `http://mriquestions.com/uploads/3/4/5/7/34572113/introduction_to_nmr_mri_amplifiers___cpc_amps.pdf` (2021).

8. Wenzel, K. *Development, evaluation and implementation of passive B0 field shimming techniques for magnetic resonance imaging* (Technische Universität Berlin, **november** 2020).

9. Anand, S. *OCRA: A Low-Cost, Open-Source FPGA-Based MRI Console Capable of Real-Time Control* (Massachusetts Institute of Technology, **september** 2018).

10. *SDRlab 122-16 Standard Kit* `https://www.redpitaya.com/p52/sdrlab-122-16-standard-kit` (2021).

11. *Opensourceimaging/Cosi-Transmit* GitHub. `https://github.com/opensourceimaging/cosi-transmit` (2021).

12. *Menkueclab/GPA-FHDO* GitHub. `https://github.com/menkueclab/GPA-FHDO` (2021).

13. *OCRA MRI - Welcome* `https://openmri.github.io/ocra/` (2021).

14. Layton, K. J. **andothers**. Pulseq: A Rapid and Hardware-Independent Pulse Sequence Prototyping Framework. *Magn Reson Med* **77** (**april** 2017).

15. Ravi, K., Geethanath, S. & Vaughan, J. PyPulseq: A Python Package for MRI Pulse Sequence Design. *JOSS* **4,** 1725 (**october** 2019).

16. Craven-Brightman, L. *lcbMGH/Ocra-Pulseq* 9 **december** 2020. `https://github.com/lcbMGH/ocra-pulseq` (2020).

17. Negnevitsky, V. *Vnegnev/Marcos_client* 24 **july** 2020. `https://github.com/vnegnev/marcos_client` (2020).

18. *OCRA MRI - Programming* `https://openmri.github.io/ocra/programming` (2021).

19. *Open File Format for MR Sequences* (). `https://pulseq.github.io/specification.pdf` (2021).

20. *Vnegnev/Marcos_extras* GitHub. `https://github.com/vnegnev/marcos_extras` (2020).

21. *URSP X300 - Overviews* `https://www.ettus.com/all-products/x300-kit/`.

22. Hasselwander, C. J., Cao, Z. & Grissom, W. A. Gr-MRI: A Software Package for Magnetic Resonance Imaging Using Software Defined Radios. *Journal of Magnetic Resonance* **270,** 47–55 (**september** 2016).

23. *Hardware:RF - Tabletop MRI* `https://tabletop.martinos.org/index.php?title=Hardware:RF` (2021).

24. *Current Driver for Local B0-Shim Coils – Open Source Imaging* `https://www.opensourceimaging.org/project/current-driver-for-local-b0-shim-coils/` (2021).