Alexander Kerschhofer, BSc

Laser Driver and Lock-In Amplifier Circuit Development for a Nitrogen Dioxide Sensor based on Quartz-Enhanced Photoacoustic Spectroscopy

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieurin

Master's degree programme: Electrical Engineering

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Mag.rer.nat. Dr.rer.nat. Alexander Bergmann

Institute of Electronic Sensor Systems

Co-Supervisor

Dipl.-Ing. Philipp Breitegger, BSc

Graz, May 2019

Acknowledgement

I want to express my great gratitude to my supervisor and head of the institute *Alexander Bergmann* for the opportunity to conduct this thesis at Institute of Electronic Sensor Systems, for providing the laboratory and work space and the financial support for this project. It was a pleasure for me to join the research team and I am grateful for your support and your patience.

A special thanks is directed to my co-supervisor *Philipp Breitegger* for the initiation of this project and the previous work on this topic, which I could base on. Your physical and technical expertise was a great support and I appreciated your cooperation in the development process and laboratory work. Furthermore, your assistance in scientific research and authoring of academic papers was of big value for me.

Additionally, I want to thank all members of the institute for the worthwhile corporation and the pleasant company.

And of course, a warm thanks to my family and friends for the encouragement during my study and the support in any situation of my life.

Abstract

Ambient air pollution is an increasing issue worldwide and a major threat to health, but also to earth's climate and the global ecosystem. The World Health Organization WHO states, that per year several million deaths can be related to air pollution, caused by diseases such as stroke, heart disease and lung cancer. The majority of the world's population suffer from poor air quality levels beyond the WHO's limits. At present, there is a rising effort to monitor air pollutants to comply with the legally prescribed concentration guidelines, which are, as an example, governed by the EU Directive for the European area. However, the spatial resolution of the measurement stations is poorly developed and the cost of appropriate equipment quite high, which inhibits the acquisition of more stations. Since low-cost sensor options lack in accuracy, the development of a new pollution sensor, in this case for nitrogen dioxide (NO_2), was launched and is the subject of discussion in this work.

The physical sensor principle is based on quartz-enhanced photoacoustic spectroscopy (QEPAS) and applies photoacoustic spectroscopy using a quartz-tuning fork (QTF) as acoustic transducer for a measurement system with a higher sensitivity of NO_2 and a lower cross-sensitivity to other air compounds. Therefore, a modulated laser light source with a specific wavelength of 450 nm excites the NO_2 molecules due to its strong absorption in this region and cause a periodic, local pressure increase by the inter-molecular relaxation processes. The resulting pressure or acoustic wave is detected by the QTF for a modulation at its resonance frequency and produces a weak electric signal of some picoamperes, which is directly proportional to the incident laser power and the pollutant's concentration. The great advantage is the high quality factor (Q-factor) of the QTF and the good environmental noise immunity.

For this concentration measurement, a high pre-amplification of the QTF signal followed by an amplification with lock-in amplifier (LIA) is required to acquire an appropriate signal quality. Furthermore, the laser has to be operated at precise modulation frequencies by the use of a laser current driver. This is currently achieved by expensive equipment and makes the sensor system non-competitive as a product on the market. To realize this at lower costs, the laser driver and the LIA are intended to be replaced by custom circuits. The main focus is given to this circuit development.

The design process of the laser driver and lock-in amplifier as well as the successful implementation in the QEPAS measurement setup are presented. A high frequency resolution of the laser modulation with an accuracy of less than 1 Hz and a controllable laser current amplitude up to 100 mA in steps of <1 mA is achieved. By the concept of an analog dual-phase LIA, the amplitude and phase information of the QTF signal can be recovered and measured with a high-resolution analog-to-digital converter (ADC). A microcontroller enables the control of the circuits during the measurement and the communication of the measured data with the PC. For the nitrogen dioxide measurements, different employed concentrations were measured and the limit of detection (LOD) determined to 784 ppb for a 10 s integration time. For a further increase in performance, some improvements are discussed. The future goal is to deploy the circuit in a prototype of a complete NO_2 sensor solution.

Zusammenfassung

Luftverschmutzung ist derzeit eine zunehmende Problematik weltweit und eine erhebliche Bedrohung für die Gesundheit, wie auch für das Klima und das globale Ökosystem. Die Weltgesundheitsorganisation WHO gibt an, dass jährlich mehrere Millionen Tode aufgrund von Erkrankungen wie Schlaganfällen, Herzerkrankungen und Lungenkrebs mit Luftverschmutzung in Verbindung stehen. Ein Großteil der Weltbevölkerung leidet dabei unter der schlechter Luftqualität mit WHO Grenzwertüberschreitung. Derzeit wird immer mehr Aufwand betrieben die Schadstoffe in der Luft zu überwachen, um die gesetzlich vorgeschriebenen Konzentrationsrichtlinien zu erfüllen, die am Beispiel von Europa durch die EU-Richtlinien vorgeschrieben werden. Dennoch ist die räumliche Auflösung der Messstationen nur spärlich ausgebaut und die Kosten des dafür notwendigen Messequipments sehr hoch, was den Ausbau des Sensor-Netzwerks verlangsamt. Da es den alternativen, preislich günstigeren Sensor-Optionen an Genauigkeit fehlt, wurde die Entwicklung eines neuen Sensors, für in diesem Fall Stickstoffdioxid (NO_2), gestartet und wird Diskussionsthema dieser Arbeit sein.

Das physikalische Sensorprinzip basiert auf der Quartz-Enhanced Photoacoustic Spectroscopy (QEPAS) und wendet die Methode der Photoakustischen Spektroskopie unter Verwendung einer Quartz-Stimmgabel (QTF) als Messwandler an, wodurch eine höhere Empfindlichkeit für NO_2 und eine geringere Querempfindlichkeit gegenüber anderen Luft Bestandteilen erreicht werden kann. Dafür regt eine modulierte Laser-Quelle mit einer Wellenlänge von 450 nm die NO_2 Moleküle an, die eine starken Absorption in diesem Lichtbereich aufweisen, und generiert eine periodische, lokale Druckerhöhung durch die intermolekularen Relaxationsprozesse. Die resultierende Druck- bzw. akustische Welle wird von der Stimmgabel bei der QTF-Resonanzfrequenz aufgenommen und erzeugt ein elektrisches Signal im Picoampere Bereich, welches direkt proportional zur eingestrahlten Laserleistung und zur Konzentration des Gases ist. Der große Vorteil hierbei ist die hohe Güte (Q-Faktor) der QTF und deren gute Störfestigkeit gegenüber akustischen Rauschquellen.

Für die Konzentrationsmessung ist jedoch eine hohe Vorverstärkung des QTF Signals und eine weitere Verstärkung mittels Lock-In Verstärker (LIA) erforderlich, um eine brauchbare Signalqualität zu erlangen. Darüber hinaus wird für die Ansteuerung des Lasers bei exakten Modulationsfrequenzen ein Lasertreiber benötigt. Dies wird aktuell mit kostenaufwendigem Equipment bewerkstelligt und macht den Sensor damit als Produkt wettbewerbsunfähig. Um die Kosten zu senken, sollen der Lasertreiber und LIA durch eine selbstentwickelte Schaltung ersetzt werden. Der Schwerpunkt dieser Arbeit liegt deshalb bei der Schaltungsentwicklung.

Der Design Prozess der Lasertreiber- und LIA-Schaltung wie auch die erfolgreiche Implementierung in den QEPAS Messaufbau werden präsentiert. Es konnte eine hohe Frequenzauflösung der Laser-Modulation mit einer Genauigkeit von weniger als 1 Hz bei einem einstellbaren Laserstrom von bis zu 100 mA in Schritten von <1 mA erreicht werden. Basierend auf dem Prinzip eines analogen Dual-Phase LIA ist die Amplituden und Phasen Information des QTF Signals ermittelbar und kann durch einen hochauflösenden Analog-Digital-Wandler (ADC) gemessen werden. Ein Mikrocontroller ermöglicht dabei die Ansteuerung der Schaltung während der Messung und die Kommunikation der Messdaten an den PC. In der Stickstoffdioxid Messung wurden verschiedene Konzentrationen aufgenommen und ein Nachweisgrenze (Limit of Detection - LOD) von 784 ppb für eine Integrationszeit von 10 s ermittelt werden. Für weitere Performance-Steigerungen werden einige Verbesserungsvorschläge betrachtet und diskutiert. Das künftige Ziel ist es, die Schaltung in einem Prototypen eines vollständigen NO_2 Sensorsystems einzusetzen.

Contents

CHAPTER 1

Introduction

Air quality is an important health factor in our society and is more and more burdened by anthropogenic air pollution with a significant contribution of industry, all kinds of traffic and wherever combustion engines are employed. Besides substances such as carbon dioxide (CO_2), sulphur oxides (SO_3), ozone (O_3) and particle matter (PM), nitrogen oxides (NO_x) and in particular nitrogen dioxide (NO_2) is one of the main air pollutants. This toxic gas bares serious health risks and is hence restricted to certain concentration limits by the EU Directive within the whole of Europe. To monitor the concentration several measuring stations are placed, especially in larger cities. At present however, the spatial resolution of these air quality measurements is poorly developed due to the cost-intensive equipment, whereby the distribution of air pollution is highly time- and location-dependent considering e.g. the evening rush hour along larger roads in comparison to low-traffic park areas.

The currently available low-cost NO_2 sensors lack in accuracy, long-time stability and cross-sensitivity to other gases and are for this reason not used for professional air quality measurements. Therefore the Institute of Electric Sensor System at the TU Graz has started the development of a NO_2 sensor based on quartz-enhanced photoacoustic spectroscopy (QEPAS), which should provide the necessary sensitivity and reduce the costs of a prospective sensor equipment. The previous work of Philipp Breitegger has proven the fundamental functionality of this QEPAS sensor [1], which this thesis is based on.

1.1 Motivation

Photoacoustic spectroscopy is a powerful technique to detect concentrations of a substance and has found its application especially in the analysis of gases. In most cases, the excitation of the gas is accomplished by an intense laser light irradiation, that is modulated at a certain frequency. The absorbed light energy causes the gas molecules to periodically expand and hence an acoustic wave. An acoustic transducer such as a microphone is measuring this signal, which is directly proportional to the incident light power and the concentration of the gas. By the choice of a specific laser wavelength, where the observed gas has a high absorption and a low cross-sensitivity to other interfering substances, the selectivity of the analyte can be defined. The quartz-enhanced photoacoustic spectroscopy (QEPAS) replaces the microphone by a quartz tuning fork (QTF) as acoustic resonator with an high quality factor (Q-factor) and offers more sensitivity at a better environmental noise immunity. This approach is the research topic of Philipp Breitegger's work for the detection of NO_2 concentrations. The concept he established requires a controllable waveform generator, a laser current driver, an amplifier circuit for the weak QTF signal as well as a lock-in amplifier for the signal detection and some optical components for the measurement setup. For the waveform generation and the lock-in amplification a PXI measurement unit from National Instruments, which is situated in a price range of a several thousand euros, is used and implemented in LabVIEW. The laser driver is an off-the-shelf component and also has an expense of some hundreds, up to thousands of euros. For a commercial product, this NO_2 measurement system exceeds the costs for an offering on the market. Hence, the goal of this master's thesis is to develop a custom circuitry for the laser driver including the frequency generation and the lock-in amplifier, in order to push down the overall costs.

1.2 Definition of Tasks

The tasks for this work are, on the one hand, the theoretical research as well as investigations on the feasibility of the NO_2 measurement with QEPAS and, on the other hand, the circuit development. In the theory part, the fundamentals of photoacoustic spectroscopy and the basic working principle of a lock-in amplifier are studied. Furthermore, the photochemistry of nitrogen dioxide with regards to photoacoustic spectroscopy, cross-sensitivity and other influences for the concentration measurement are examined.

The circuit development is split into the two parts, laser driver and lock-in amplifier (LIA). For the laser driver it is specified to have a high frequency resolution of less than 1 Hz in the range of 32 to 33 kHz for the laser modulation and a current output adapted to the requirements of the applied laser diode. As reference signals for the LIA an additional sine and cosine (90° phase-shifted, quadrature signal) output with the same frequency and phase as the modulation has to be provided.

The LIA circuit is intended for the weak QTF signal detection and has to recover the magnitude and phase information, which can be realized i.a. with a dual-phase LIA. To measure the lock-in amplifier output, the result should be converted with a high-resolution analog-to-digital converter (ADC). Both circuits should be controllable by a microcontroller or via the PC and communicate the resulting data for post-processing.

CHAPTER 2

Theory

This chapter is devoted to the theoretical background of photoacoustic spectroscopy, the functional principle of a lock-in amplifier and some fundamentals on the physics of nitrogen dioxide NO_2. The first part gives an introduction to the basic of absorption spectroscopy, the relation between intensity and concentration and the common realizations of photoacoustic spectroscopy (PAS). The second part explains the main functionality of lock-in amplifiers, which is common equipment for the photoacoustic signal analysis. In the third part the possibilities and requirements of measuring NO_2 by means of photoacoustic spectroscopy are discussed.

2.1 Photoacoustic Spectroscopy

Many techniques for the spectroscopic analysis of gases are based on optical absorption. Among them, the three established methods, that are discussed in the following sections, are absorption spectroscopy, photoacoustic spectroscopy (PAS) and quartz-enhanced photoacoustic spectroscopy (QEPAS), which are commonly used in gas concentration experiments. All of them refer to the absorption wavelength and absorption strength of the medium [2]. For PAS the physical principle is the photoacoustic (PA) effect, which explains the acoustic wave generation after modulated excitation of the medium [3]. Additional improvement in sensitivity for certain application can be achieved with QEPAS by utilizing a quartz tuning fork (QTF) as transducer [4].

2.1.1 Absorption Spectroscopy

Absorption spectroscopy is an analytical method based on the interaction between matter and radiation. It measures the absorption of radiation on the transmission path trough the observed medium and can be applied in the whole range of electromagnetic radiation. The characteristic absorption spectrum of a material represents the variation of absorbed energy over a range of frequencies. In the case of light radiation photons or rather their energy is absorbed and the absorption spectrum is often expressed as function of wavelength or wavenumber. Because the ability to absorb light is linked to the quantum mechanical states of a molecule, absorption appears more likely at frequencies where the energy match the difference of two states. This is referred to as absorption lines in a spectrum [5, 6].

Absorption and Emission

Caused by the absorption of a photon the internal energy of a molecule is increased. This leads to rotational, vibrational and electronic excited energy states depending on the photon energy. Rotational excitation already occurs at low energies in the microwave to far infrared (FIR) range, vibrational states can be stimulated by infrared/near-infrared (IR, NIR) light and electronic states by visible to ultra-violet (UV) light. However, exitation to a vibrational level is generally combined with an increase of rotational energy and an electric exitaion with a change in vibrational and rotational energy levels [7]. To return to its ground state the molecule releases the energy of the excited states by radiative or non-radiative processes.

Jablonski Diagram

The Jablonski diagram (2.1) is a simplified model of the electric energy levels and the transitions between them. The involved states are grouped by their spin multiplicity. States with a multiplicity $(2S + 1) = 1$ are called singlet and indicate all spins anti-parallel while states with a multiplicity $(2S + 1) = 3$ called triplet and have two parallel spins. Here the mentioned S corresponds to the total spin angular momentum. The ground state is annotaded as S_0, the other singlet and triplet states as S_n and T_n. Due to the forbidden intercombinations, absorption is only allowed within the same multiplicity. Excitation occurs from S_0 to one of the higher-energetic singlet states S_n [7, 8].

Considering only the first, optical excitable singlet state S_1 and the slightly lower triplet state T_1, Figure 2.1 illustrates the possible transitions. Besides the two types of radiative processes/emission ($\rightarrow$), there are non-radiative processes possible ($\rightsquigarrow$). The first emission process is the fluorescence from ($S_1 \rightarrow S_0$) and emits light with an energy that is equal to the difference of the energy levels of S_1 and S_0. The second is called phosphorescence ($T_1 \rightarrow S_0$) and appears after the non-radiative relaxation of intersystem crossing ($S_1 \rightsquigarrow T_1$). Compared to fluorescence this is a much longer process caused by the time span of the intersystem crossing and the longer lifetimes in the triplet state T_1 of some milliseconds to minutes. Phosphorescence is a weak, long-lasting radiation with a smaller energy (larger wavelength) due to the Stokes shift[1]. Alternatively deactivation of the excited singlet S_1 can take place by non-radiative, vibrational-rotational relaxation either directly from $S_1 \rightsquigarrow S_0$ or after intersystem crossing from $T_1 \rightsquigarrow S_0$ [7, 8]. Thereby the absorbed energy is released as translational, kinetic energy and furthermore as heat [3].

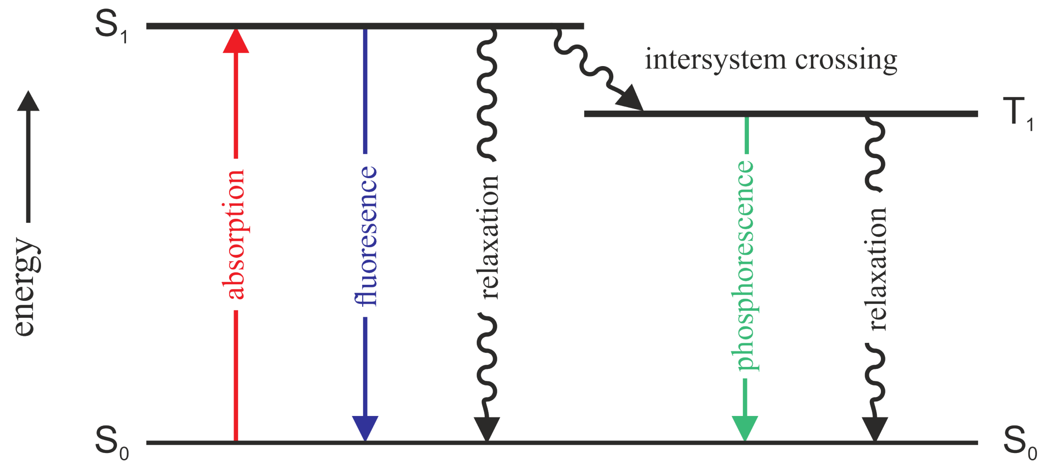


Figure 2.1: Jablonski diagram showing the first singlet S_1 and triplet T_1 states and the possible relaxations after optical excitation from the ground state S_0.

Quantitative Measure of Absorption

For quantitative measurements of gaseous species using absorption spectroscopy it is important to discuss the mathematical background of absorption. The measurable value of absorption is the intensity of the transmitted light along its path through the medium.

[1] In addition to the Stokes shift, materials can have the ability to absorb a second photon out of the first triplet state and reach a doubly excited state. This is known as anti-Stokes shift and shifts the emission spectrum to higher energies (smaller wavelengths). An example for such a material are the multiphoton phosphors [9].

Based on the Beer-Lambert law, the wavelength-dependent transmitted light intensity $I_t(\lambda)$ is determined by

$$I_t(\lambda) = I_i \cdot e^{-\tau} = I_i \cdot 10^{-A}, \tag{2.1}$$

where λ is the wavelength, I_i the incident light intensity, τ the optical depth and A the absorbance of the medium [2, 10]. In Equation (2.2) the relation between wavelength λ, wavenumber $\tilde{\nu}$ and the light-wave frequency ν is given as

$$\tilde{\nu} = \frac{1}{\lambda} = \frac{\nu}{c} = \frac{\Delta E}{h}, \tag{2.2}$$

in which c is the speed of light, ΔE the energy difference between the upper and lower energy level and h the Planck's constant [2].

Absorbance is the quantitative measure of attenuation. In addition to absorption, attenuation includes effects of reflection, scattering and luminescence. Neglecting these effects the optical depth τ often is used as the term absorption a. The absorbance A (Eq. (2.3)) is defined as the dimensionless, common logarithm of incident to transmitted light, whereas the optical depth τ (Eq. (2.4)) is defined as the natural logarithm.

$$A = log_{10}\left(\frac{I_i}{I_t}\right) \tag{2.3}$$

$$\tau = ln\left(\frac{I_i}{I_t}\right) \tag{2.4}$$

Provided that the attenuation in the material is uniform, both are related to the attenuation coefficient $\mu(\lambda)$ in $\mathrm{cm^{-1}}$ by the path length l in cm. The relation to molar attenuation coefficient $\epsilon(\lambda)$ in $\mathrm{cm^2\,mol^{-1}}$ (per mole) and attenuation cross section $\sigma(\lambda)$ in $\mathrm{cm^2}$ (per molecule) is given as follows

$$A = \mu_{10}(\lambda) \cdot l = \epsilon(\lambda) \cdot c \cdot l \tag{2.5}$$

$$\tau = \mu(\lambda) \cdot l = \sigma(\lambda) \cdot n \cdot l, \tag{2.6}$$

where the decadic attenuation coefficient μ_{10} is the attenuation coefficient μ devided by $ln(10)$, c the molar concentration $\mathrm{mol\,cm^{-3}}$ and n the number density in $\mathrm{cm^{-3}}$ [10]. The term absorption coefficient α can be set equal to the attenuation coefficient μ in the case of negligible small effects of reflection, scattering and luminescence.

By definition of the Avogadro constant N_A, which represents the number of molecules contained in an amount of substance of one mole, the molar concentration c is linked to the number density n.

$$n = N_A \cdot c, \quad N_A = 6.022140857 \cdot 10^{23}\, mol^{-1} \tag{2.7}$$

In atmospheric science it is common to use the dimensionless volume fraction Φ_i of a species i in percent % or parts-per notation for the concentration of gases, e.g. air pollutants (Equation (2.9)). To calculate this volume fraction, the total number density N_{tot} has to be known. This is defined as the number of molecules per volume V and given by the ideal gas law in Equation (2.8)

$$N_{tot} = \frac{p \cdot V}{k_B \cdot T}, \tag{2.8}$$

at a certain pressure p in $Pascal$ and temperature T in K, where k_B is the Boltzmann constant. For the calculation of the volume fraction, N_{tot} has to be calculated to the same volume as the number density n_i of the species i (usually cm^{-3}).

$$\Phi_i = \frac{n_i}{N_{tot}} \quad \leftrightarrow \quad n_i = \Phi_i \cdot N_{tot} \tag{2.9}$$

To convert volume fraction into number density this formula can be used vice versa.

2.1.2 Photoacoustic Effect

The photoacoustic (PA) effect is the acoustic wave generation of molecules induced by optical absorption processes. Figure 2.2 illustrate these physical processes of the PA effect. As discussed in Section 2.1, gas molecules can absorb photons, get into an excited energy state and release this energy in radiative or non-radiative processes. For the PA effect only the non-radiative relaxation processes of vibration-vibration (V-V) and vibration-rotation, translation (V-R, T) are important. These achieve a local heat generation by kinetic energy release, that results in a local pressure increase. When the incident light intensity is modulated, the heat generation will be periodic and occurs as pressure wave with the same frequency as the modulation frequency. For modulation frequency in the acoustic range this wave appears as acoustic wave. The acoustic wave can further be detected by an acoustic transducer, such as a microphone [3–5].

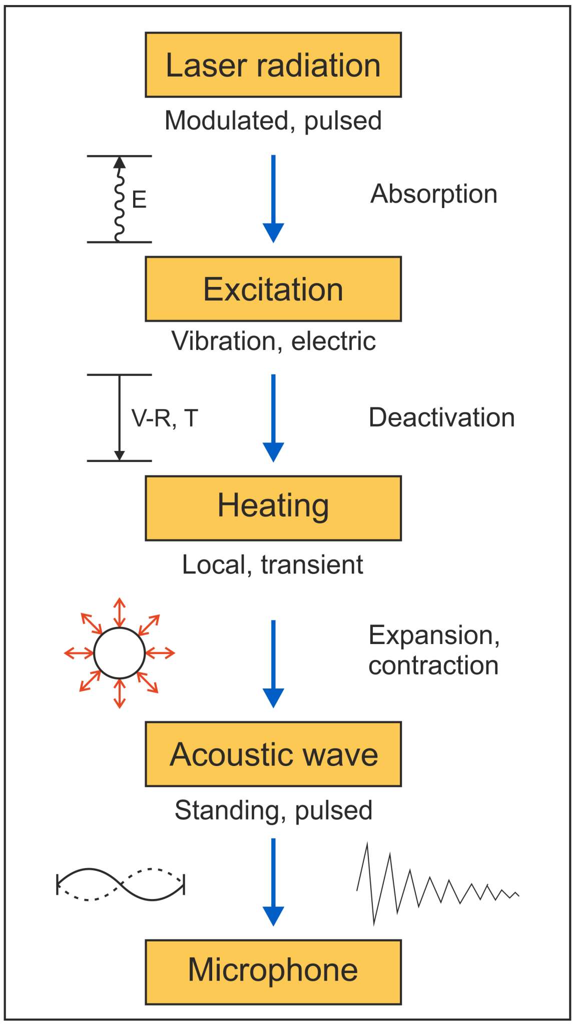


Figure 2.2: Detailed sheme of photoacoustic relaxation processes: Rotational, vibrational and electronic excited states deactivate by R-T; V-R, T and E-V, R, T processes and generate local transient heating. The resulting expansion is detectable as standing or pulsed acoustic wave by e.g. a microphone [3]

The major part of the photoacoustic signal is generated by the V-R, T relaxation. For vibrational exitation states the radiative emission (phosphorescence) is of minor importance and the relaxation time much longer [5]. Typical relaxation times at atmospheric conditions for vibrational relaxations are in the range of 10^{-5} to 10^{-9} s whereas radiative relaxation is in the range of 10^{-1} to 10^{-3} s [5, 11].

One big advantage of photoacoustic spectroscopy, as compared to conventional absorption spectroscopy, is the linear relation between concentration and measured, photoacoustic signal $S(\lambda)$. This signal will be proportional to the attenuation coefficent $\mu(\lambda)$, the number density and the incident light power $P_i(\lambda)$ for the certain wavelength λ.

This relation is given in Equation (2.10)

$$S(\lambda) = C \cdot P_i(\lambda) \cdot \mu(\lambda) = C \cdot P_i(\lambda) \cdot \sigma(\lambda) \cdot n, \tag{2.10}$$

where C is the instrumental constant based on the experimental setup, $\sigma(\lambda)$ the attenuation cross section and n the number density [4]. For calculation of molar concentration c or dimensionless concentration in volume fraction Φ confer on equations (2.7) and (2.9).

2.1.3 Quartz-Enhanced Photoacoustic Spectroscopy

Quartz-enhanced photoacoustic spectroscopy (QEPAS) is a specialized method of photoacoustic spectroscopy (PAS). Its fundamental physical effect stays the same, but only the acoustic transducer is replaced by a quartz tuning fork (QTF). This tuning fork is a piezoelectric quartz crystal and has a sharp resonance with a width of $f_{FWHM} \approx 4\,\text{Hz}$. High-Q (high quality factor) quartz crystals are applied in watches or other electric devices as frequency standard and therefore are mass-produced and low-cost. The most common QTF, as it is pictured in Figure 2.3, has a resonant frequency of around $2^{15}\,\text{Hz}$ or 32.768 kHz, but it can be designed to any custom frequency between 4-200 kHz as well [4, 12]. The mode of the tuning fork at resonance corresponds to a symmetric vibration, where the prongs oscillate in opposite directions. Antisymmetric vibration is piezoelectrically not possible, which provides a good environmental noise immunity. Hence, only acoustic waves with frequencies in the narrow band of resonance, that are between this prongs can produce excitation of the QTFs vibration. In addition, a typical watch QTF assure a high Q-factor of $Q \approx 20\,000$ in encapsulated condition and around $Q \approx 8000$ at atmospheric pressure [12]. In conclusion, all the listed properties constitute the quartz tuning fork as excellent, acoustic transducer for photoacoustic spectroscopy.

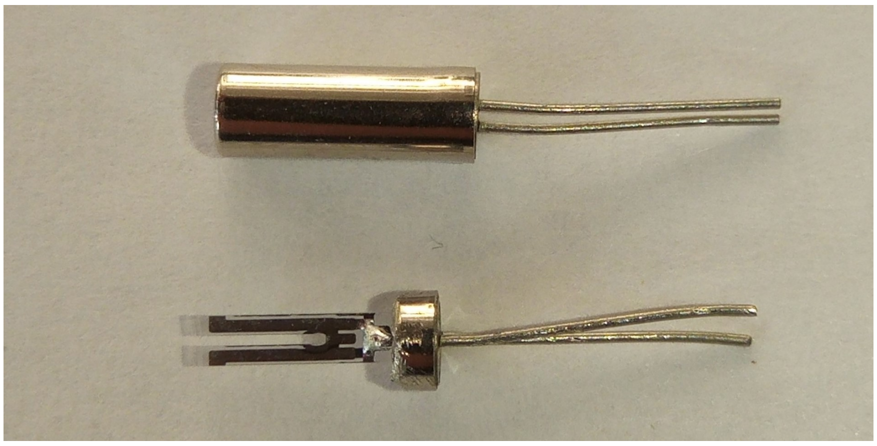


Figure 2.3: Quartz tuning fork in encapsulated form and with removed cap

QEPAS measurements are performed with a focused laser beam pointing through the tuning fork at modulation frequencies around the resonance of the QTF ($\sim 32\,\mathrm{kHz}$). This rate of periodic excitation is about the relaxation time of V-R, T relaxation processes in gases and thus the sensitivity is even higher compared to the PAS, that is typically driven at lower frequencies. Only for molecules with a very slow translational relaxation rate the detection gets weaker, since the laser modulation is then too fast for the molecular energy transfer. In the other case, where the V-R, T relaxation is moderate fast, the photoacoustic signal $S(\lambda)$ can be expressed similar to the PAS [4, 12].

$$S(\lambda) \propto \frac{Q \cdot P_i(\lambda) \cdot \mu(\lambda)}{f_0}, \tag{2.11}$$

with Q representing the quality factor and f_0 the resonant frequency. The Q-factor for the QTF is typically in the range of 10^5, but can also experimentally be determined by measuring the resonant frequency f_0 and the full width at half maximum Δf_{FWHM}. Equation (2.12) allows the calculation of Q [4, 12].

$$Q = \frac{f_0}{\Delta f_{FWHM}} \tag{2.12}$$

Figure 2.4 shows an overview of an typical QEPAS setup for gas concentration measurements. A function generator produces the modulation signal and is applied to a laser driver, which controls the laser diode current. The laser beam is collimated and focused towards the QTF by lenses. If the laser is driven at the QTF's resonant frequency f_0 and the investigated gas is present, which is photoacoustically active at the wavelength λ, the tuning fork will output a weak electric signal. The QTF signal is passed through a pre-amplifier working in transimpedance configuration to acquire a low-noise signal with proper amplitude. Further on, this is measured using a lock-in amplifier[2] with the modulation frequency as reference to obtain only that frequency component and to suppress the noise. [12, 14].

[2] Lock-in amplification (LIA) is a technique to extract signals with a known frequency, that have a noisy background. The frequency and phase-sensitive methode is perfect for recording weak signals, because of its high detection accuracy, and is widely used in the field of absorption spectroscopy [13] (more details in Section 2.2).

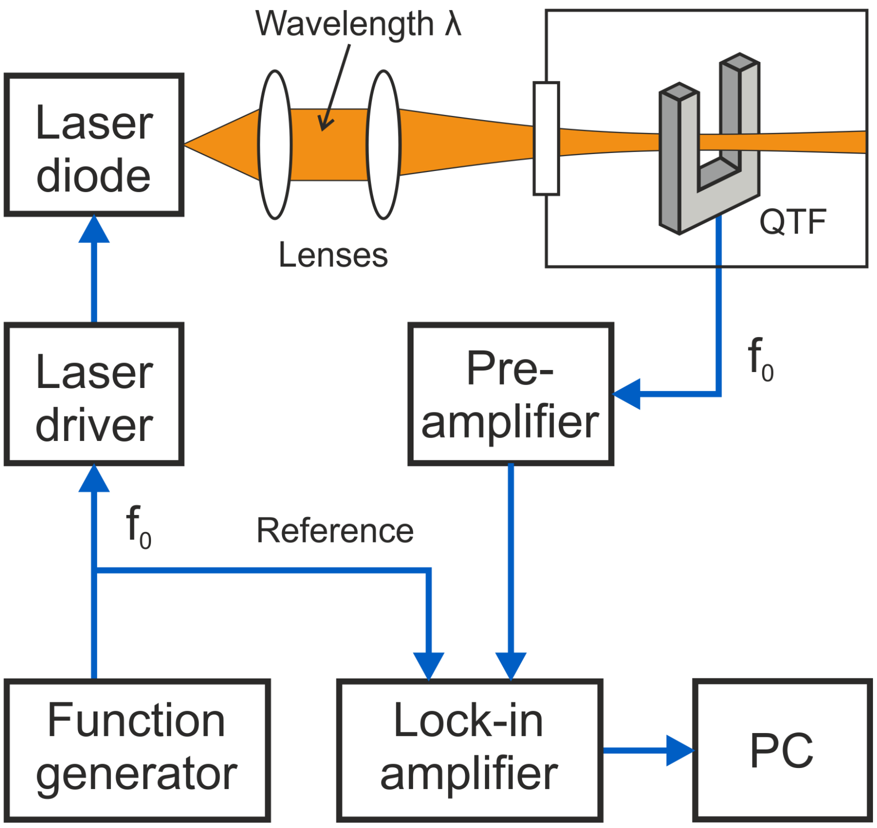


Figure 2.4: Experimental setup for QEPAS measurements: The function generator produces the control/reference signal at the QTF's resonant frequency f_0. The laser driver converts this into a current for the laser, which is focused between the tuning fork's prongs. The QTF output is pre-amplified and measured by the lock-in amplifier connected to a PC.

2.2 Lock-In Amplifier

A Lock-in amplifier (LIA) is an accurate measurement instrument capable of detecting very small AC signals, even in presence of extremely large noise signals up to a million times higher in amplitude. Modern lock-in amplifiers have been developed to instruments offering a variety of functions. Their scientific purposes provide efficient solutions for a huge number of measurement applications. In fields of research, where high accuracy and signal-to-noise ratio (SNR) are required, they became an important measurement equipment in all kinds of laboratory setups. Examples for the most prominent use cases are listed below [15, 16].

- AC signal recovery
- Weak AC signal detection
- Precicion AC voltage and phase meter
- Noise measurement unit
- Spectrum analyser
- Network analyser

By means of phase-sensitive detection and low-pass filtering the LIA extracts the signal component at a specific reference frequency and rejects all other frequency components. The provided DC output is proportional to the AC signal's amplitude and phase. Lock-in amplification can be realized in different ways, but the main working principle of the phase-sensitive detector (PSD) stays the same. The traditional analog LIA operates only with analog input and reference signals, whereas modern devices use an analog-to-digital converter (ADC) to immediately convert the input signal into a digital representation and perform the steps of demodulation and filtering mathematically by fast digital signal processing (DSP), e.g. on field programmable gate arrays (FPGA). Additionally to the analog output signal, latter ones usually provide the DC output as value on a display and/or as a digital value available on the PC, communicated over an interface port [15–17].

2.2.1 Phase-Sensitive Detection

Phase-sensitive detection, as it is performed in lock-in amplifiers (LIA), is basically the multiplication of the input with a reference signal, that has the same frequency as the observed signal. This reference can either be generated by the LIA itself or has to be provided externally. Most common waveforms for the frequency reference are sine wave or square wave. The phase-sensitive detector (PSD), which is also referred to as demodulator or mixer, operates the multiplication of both signals and the low-pass filtering is accomplished with a first or higher order RC filter in an analog LIA or by mathematical multiplication on the DSP for a digital LIA. [15–17].

To demonstrate this mathematically and graphically, assume the signal as sine wave $V_s(t)$, where A_s is the signal amplitude, ω_s the circular signal frequency and Θ_s is the phase of the signal. The sinusoidal reference is $V_r(t)$ with the amplitude A_r, the circular frequency ω_r and phase Θ_r. After demodulation the PSD output results to $V_{PSD}(t)$ [16, 17].

$$\begin{gathered} V_s(t) = A_s \cdot sin(\omega_s t + \Theta_s) \quad \text{and} \quad V_r(t) = A_r \cdot sin(\omega_r t + \Theta_r), \\ \text{multiplies to} \\ V_{PSD}(t) = A_s A_r \cdot sin(\omega_s t + \Theta_s) \cdot sin(\omega_r \cdot t + \Theta_r) \\ = \tfrac{1}{2} A_s A_r \cdot cos([\omega_s - \omega_r]t + \Theta_s - \Theta_r) - \tfrac{1}{2} A_s A_r \cdot cos([\omega_s + \omega_r]t + \Theta_s + \Theta_r) \end{gathered} \tag{2.13}$$

The outcome of the PSD is a superposition of these sinusoidal signals with the high and low frequency components $(\omega_s + \omega_r)$ and $(\omega_s - \omega_r)$. When a low-pass filtering is applied to $V_{PSB}(t)$, the high frequeny component is removed. Consequently the low frequency output remains. If the signal and reference frequencies are equal, i.e. $\omega_s = \omega_r$, the AC signal $V_{PSD,\,AC}$ (Eq. (2.14)) has twice the frequency. The DC component $V_{PSD,\,DC}$ (Eq. (2.15)), that is available behind the low-pass filter (LP filter), then corresponds to the mean value of the PSD signal. This is proportional to the amplitudes A_s and A_r (this one is known and constant) and related to the phase difference from signal to reference $\Theta = \Theta_s - \Theta_r$ [16,17].

$$V_{PSD,\,AC}(t) = -\tfrac{1}{2}A_sA_r \cdot cos(2\omega_s t + \Theta_s + \Theta_r) \tag{2.14}$$

$$V_{PSD,\,DC}(t) = \tfrac{1}{2}A_sA_r \cdot cos(\underbrace{\Theta_s - \Theta_r}_{\Theta}) = \tfrac{1}{2}A_sA_r \cdot cos(\Theta) \tag{2.15}$$

The following plots in Figure 2.5 illustrate the mentioned signals before the LIA (V_s, V_r), after the PSD (V_{PSD}) and the DC component after filtering ($V_{PSD,\,DC}$) for three different cases[3]:

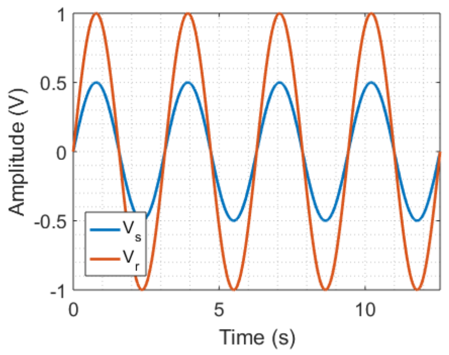
1. Input and reference signal have the the same frequency and phase
2. The signals are in-phase, but the signal has another frequency
3. The frequencies are equal, but the two signals are out-of-phase

1. Figure 2.5a shows the input signal V_s and the reference V_r with exactly the same frequency $\omega_s = \omega_r$ and zero phase shift $\Theta = 0$ over time. The signal after demodulation (V_{PSD}) in Figure 2.5b has twice the frequency and represents the sum of the AC signal $V_{PSD,\,AC}$ of Equation (2.14) and the DC offset $V_{PSD,\,DC}$ of Equation (2.15). After the LP filter only $V_{PSD,\,DC}$ remains.

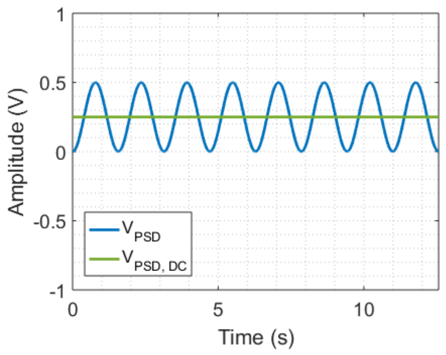
2. For the case, that the frequencies differ ($\omega_s = 2.5 \cdot \omega_r$) and the input signal has the same phase as the reference, Figure 2.5c demonstrates V_s and V_r. The output signal V_{PSD} of the PSD is an superposition of both signals with the two frequency components $\omega_s - \omega_r$ and $\omega_s + \omega_r$. This and the average signal $V_{PSD,\,DC} = 0$ is shown in Figure 2.5d.

3. In Figure 2.5e the signal V_s is phase-shifted by 45° in respect to the reference V_r, their frequencies are the same. The PSD signal V_{PSD} is similar to the first case, but only the DC component $V_{PSD,\,DC}$ got smaller by $cos(\Theta)$. V_{PSD} and the remaining signal after LP filtering $V_{PSD,DC}$ can be seen in Figure 2.5f.

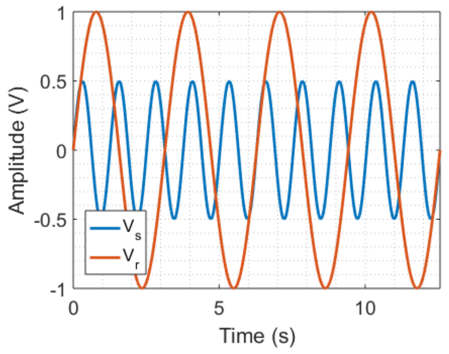
[3] The phase angle of the reference signal Θ_r is set to zero for this Illustration



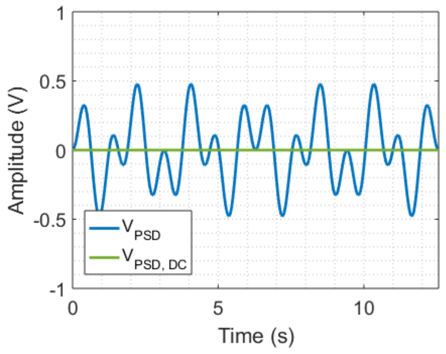
(a) The input signal V_s has the same frequency and phase as the reference V_r



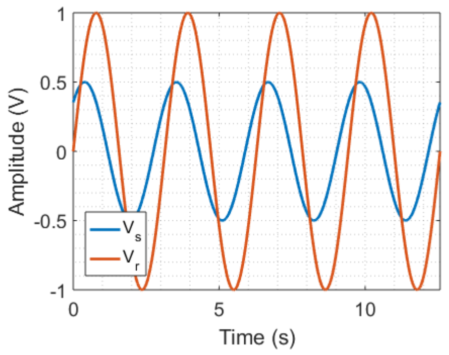
(b) The resulting signal V_{PSD} is of twice the frequency and an amplitude of $\frac{1}{2}A_sA_r$. Its mean value $V_{PSD,\ DC}$ remains after low-pass filtering.



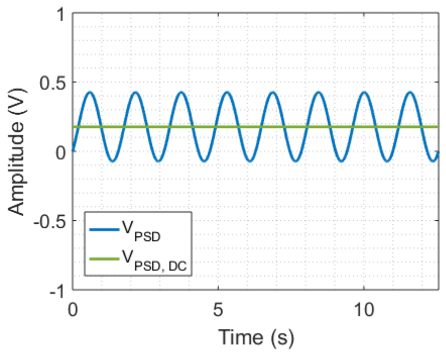
(c) The signal's frequency ω_s increased by a factor of 2.5.



(d) The resulting signal V_{PSD} has two frequency components $\omega_s - \omega_r$ and $\omega_s + \omega_r$. The mean value $V_{PSD,\ DC}$ is zero.



(e) The resulting signal V_{PSD} has a phase shift of 45° to the reference V_r.



(f) V_{PSD} has a smaller DC offset compared to **(b)**. The low-passed $V_{PSD,\ DC}$ decreased.

Figure 2.5: An input signal V_s (left plots, blue) with an amplitude of $0.5\ V$ is demodulated with the $1\ V$ reference signal V_r (left plots, red). The frequency and phase properties of the signal V_s are varied. On the right, the corresponding, resulting signals after the PSD V_{PSD} (blue) and the low-passed signals $V_{PSD,\ DC}$ (green) are illustrated.

In order to determine the amplitude of an observed signal by lock-in amplification, the frequency and the phase of the signal has to be known to generate an appropriate reference signal. This is often performed automatically by modern, high-end instruments. An other common method is to use dual-phase demodulation, which is further discussed in Section 2.2.2. As part of this, the typical LIA circuit will be explained.

2.2.2 Dual-Phase Demodulation

Dual-phase demodulation incorporates two single-phase LIAs consisting of one PSD and an LP filter, which are working in parallel. The only difference is, that the second one is supplied with a reference signal, which is phase-shifted by $90° = \frac{\pi}{2}$. The two outputs are called in-phase X and quadrature signal Y. This method is used to avoid phase shifting of the reference signal to be in-phase with the input signal, which is another approach for LIA implementations.

For a better understanding, the resulting signal after LP filtering of Equation (2.15) is represented as a phasor in the complex plane. The in-phase signal X is the multiplication of the signal $V_s(t)$ with a reference $V_r(t)$, which has a phase of $\Phi_r = 0$.

$$X = \underbrace{\tfrac{1}{2}A_sA_r}_{R_s} \cdot cos(\Theta_s - \Theta_r)\Big|_{\Theta_r=0} = R_s \cdot cos(\Theta_s) = R_s \cdot cos(\Theta_s) \tag{2.16}$$

The magnitude of the phasor is denoted as R_s and the phase as Θ_s. Is the input signal multiplied with the 90° phase-shifted reference ($\Theta_r = \frac{\pi}{2}$) the output of the LIA will be the quadrature signal Y.

$$Y = \underbrace{\tfrac{1}{2}A_sA_r}_{R_s} \cdot cos(\Theta_s - \Theta_r)\Big|_{\Theta_r=\frac{\pi}{2}} = R_s \cdot cos(\Theta_s - \tfrac{\pi}{2}) = R_s \cdot sin(\Theta_s) \tag{2.17}$$

The complex signal Z is denoted as the complex sum of X and Y, which can be further simplified by the trigonometric identity of sine and cosine (Euler's formula)

$$Z = X + jY = R_s \cdot e^{j\Theta_s}, \tag{2.18}$$

where j is the imaginary number.

By transformation from Cartesian into polar coordinates the amplitude R_s and phase Θ_s can be calculated as follows [15, 17].

$$\begin{aligned} R_s &= \sqrt{X^2 + Y^2}, \\ \Theta_s &= arctan\left(\tfrac{Y}{X}\right) \end{aligned} \tag{2.19}$$

This mathematical transformation can be visualized in a phasor diagram (Figure 2.6). Therefore, the complex signal $Z = R_s \cdot e^{j\Theta_s}$, is split into real part $X = R_s cos(\Theta_s)$ and imaginary part $Y = R_s sin(\Theta_s)$ and drawn in the complex plane.

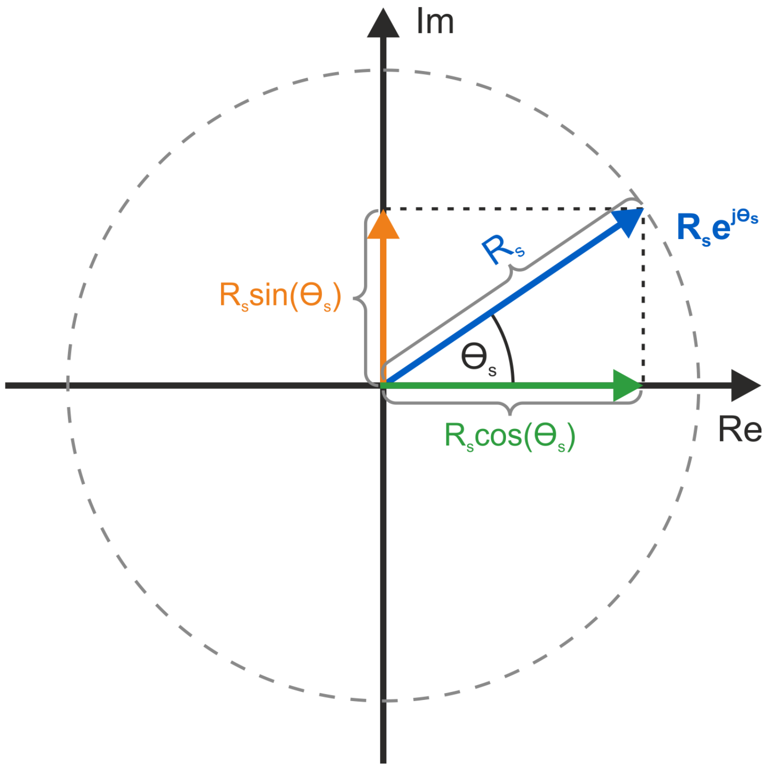


Figure 2.6: Phasor diagram of the complex signal $Z = R_s \cdot e^{j\Theta_s}$ (blue). The projection on the real axis Re represents the in-phase component $X = R_s cos(\Theta_s)$ (green) and the projection on the imaginary axis Im the quadrature component $Y = R_s sin(\Theta_s)$ (orange). The magnitude R_s of the resulting, lock-in amplified signal is given by the length of the phasor in blue, the phase Θ_s by the angle between the phasor and the real axis.

As shown above, dual-phase demodulation is one elegant method to measure the in-phase and quadrature component of a signal. Using the trigonometric relations, the signal amplitude and phase can be derived without any knowledge of the phase difference from signal to reference.

In Figure 2.7 an exemplary sketch of the dual-phase demodulation circuit (or dual-phase LIA) is shown. The input signal $V_s(t)$ is applied to the two single-phase LIAs and is separately multiplied with the reference signal $V_r(t)$ and the 90° phase-shifted copy. Afterwards both signals are passed through the LP filters and result in the two outputs of in-phase X and quadrature component Y. According to Equation (2.19), the vector computation is performed to obtain the amplitude R_s and phase Θ_s.

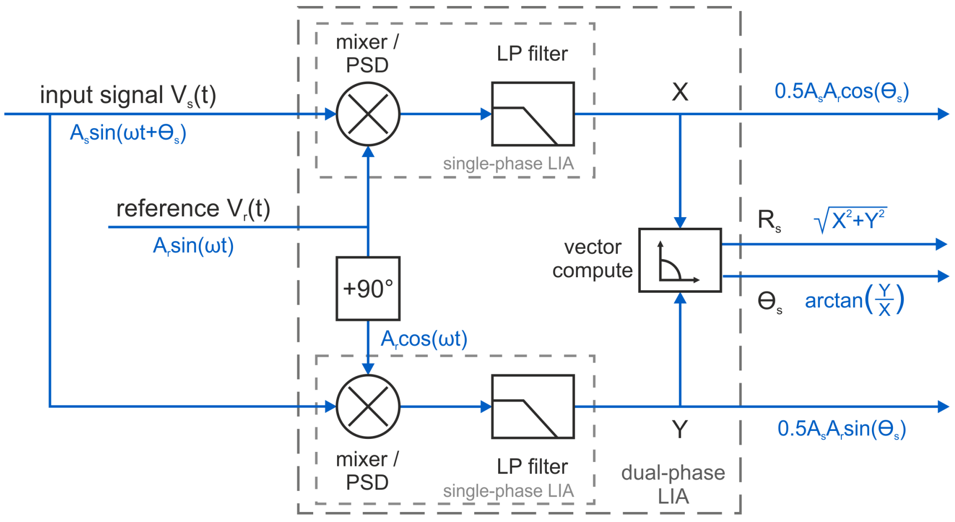


Figure 2.7: Dual-phase LIA consisting of two single-phase LIAs. The input signal $V_s(t)$ is multiplied once with the reference signal $V_r(t)$ and the 90° phase-shifted copy and low-pass filtered afterwards. The resulting outputs X (in-phase) and Y (quadrature component) are trigonometrically computed to extract the amplitude R_s and phase Θ_s information.

2.2.3 Types of Phase-Sensitive Detectors

In general there are three ways for phase-sensitive detection, that can be used in lock-in amplifiers. These are shortly explained and their advantages as well as disadvantages are discussed in the following [16, 17].

Analog Multiplier

The PSD implementation with an analog multiplier is an electric circuit, which simply multiplies the the input with the reference signal of the same frequency in the analog domain [16, 17].

+ Simple and practicable circuit
+ Large bandwidths can be achieved
− Difficulties to multiply linearly in the presence of large noise, poor dynamic reserve
− Smaller harmonic rejection, for square wave reference influence at odd harmonics
− Output offset caused by analog multiplier and gain error due to changes of the reference's amplitude

Digital Switching Multiplier

The digital switching multiplier is an analog polarity switch driven at the reference frequency. It multiplies the input signal by +1 when the reference is positive and by −1 when the reference is negative [16, 17].

+ Very simple circuit and common application in demodulators
+ Work up to high frequencies, restricted only to switching delay
+ Good linearity over a large range of input signals
+ Independent of reference amplitude, no gain error
− Detects also odd harmonics (quasi square wave), filtering required
− Noise at odd harmonics limit the dynamic range

Digital Multiplier
Multiplication in an digital LIA is completely performed on the digital signal processor (DSP). Therefore the input signal is digitalized by an analog-to-digital converter (ADC) and this digital representation multiplied by the digital sequence of the reference [16, 17].

+ Mathematically perfect multiplication
+ Analog LP filters are obsolete, performed digitally
+ Digital output value available, usually with PC interface for data processing
− Bandwidth is restricted by the sampling frequency (Shannon: $f_s/2$) of the ADC
− In presence of large noise the signal recovery is limited to the dynamic range of the ADC (typically 16-bit), ADCs with higher accuracy and fast sampling rates are extremely expensive or not available with these requirements

2.2.4 State of the Art

The progress in electronics pushed the development of ADCs with higher resolutions and speed as well as the signal processing in real time on FPGAs. Hence, state-of-the-art instruments use the digital multiplier type for their phase-sensitive detection. The additional advantage of digital data analysis in time and frequency domain is to perform different calculations at once. This for example combines the capabilities of lock-in amplifier, scope and spectrum analyser in on device. One of the advanced instruments is Zurich Instruments' UHFLI with an input bandwidth of 600 MHz (5 MHz demodulation bandwidth) and a noise performance of $4\,\mathrm{nV}/\sqrt{\mathrm{Hz}}$ at a dynamic reserve of a 100 dB [15].

2.3 Photochemistry of Nitrogen Dioxide

Nitrogen dioxide NO_2 is a reddish-brown, toxic gas and a prominent air pollutant mainly caused by traffic from vehicle engines or other types of combustions, where fuels are burned. It belongs to the nitrogen oxides (NO_x), that are one of the air pollution indicators in atmospheric chemistry.

The photochemistry of NO_2 has been well-investigated for decades and a lot of reference data, such as attenuation cross sections, are available. Nevertheless, the nitrogen dioxide's chemical properties are quite challenging for concentration measurements using photoacoustic spectroscopy. The following important aspects are further discussed in detail:

- The absorption spectrum of NO_2 and the possible cross-sensitivity with other atmospheric gases at standard air concentrations
- The photodissociation/photolysis of NO_2 molecules in the UV-VIS range
- Its equilibrium dimer N_2O_4 and the dependency on temperature and pressure

2.3.1 Absorption Spectrum

In this thesis the photoacoustic spectroscopy of NO_2 is investigated for ultraviolet and visible (UV-VIS) radiation as excitation energy. This is advantageous, because less air compounds are sensitive in this area compared to the IR (infra-red) region. Still there are some gases, that have to be considered. These atmospheric gases with absorption in the UV-VIS range are summarized in Table 2.1 including their spectral absorption wavelengths in this range and the known, typical relative concentrations (in volume fraction).

gas		absorption wavelength (nm)	approx. volume fraction
nitrogen	N_2	<100	78.08 %
oxygen	O_2	<245	20.95 %
water vapor	H_2O	<210 >600	0.1 - 7 %
ozone	O_3	170 - 350 450 - 750	30 - 50 ppb
nitrous oxide	N_2O	<240	30 ppm
nitrogen dioxide	NO_2	200 - 600	5 - 20 ppb
dinitrogen tetroxide	N_2O_4	<450	10 - 40 ppb
nitric oxide	NO	<200	n/s
nitrate radical	NO_3	410 - 670	<2 ppb
nitrous acid	HNO_2	<400	~1 ppb
nitric acid	HNO_3	<330	~0.5 ppb
methyl bromide	CH_3Br	<260	n/s
CFC11	$CFCl_3$	<230	n/s
formaldehyde	HCHO	250 - 360	n/s

Table 2.1: Atmospheric gases with absorption in the UV and visible spectrum, their absorption range and the known, approximated, typical concentrations (volume fraction) in ambient air (not specified/known indicated by n/s) [18–21].

The Max-Planck-Institute for Chemistry provides a comprehensive collection of spectral attenuation cross sections $\sigma(\lambda)$ and quantum yields $\phi(\lambda)$ of photolysis in their database "The MPI-Mainz UV/VIS Spectral Atlas of Gaseous Molecules of Atmospheric Interest" [22]. By use of that database all the important gases, which could interfere with the nitrogen dioxide absorption, were analysed and the attenuation spectra compared. The gases with attenuation cross sections interfering with that of NO_2 are plotted between 200 and 800 nm and shown in Figure 2.8. These are the nitrogen compounds dinitrogen tetroxide N_2O_4 (the dimer of NO_2), nitrate radical NO_3, nitrous acid HNO_2 and nitric acid HNO_3 as well as ozone O_3. The attenuation cross sections are referred to room temperature around 298 K at atmospheric pressure $1\,\text{atm} \approx 1\,\text{bar}$.

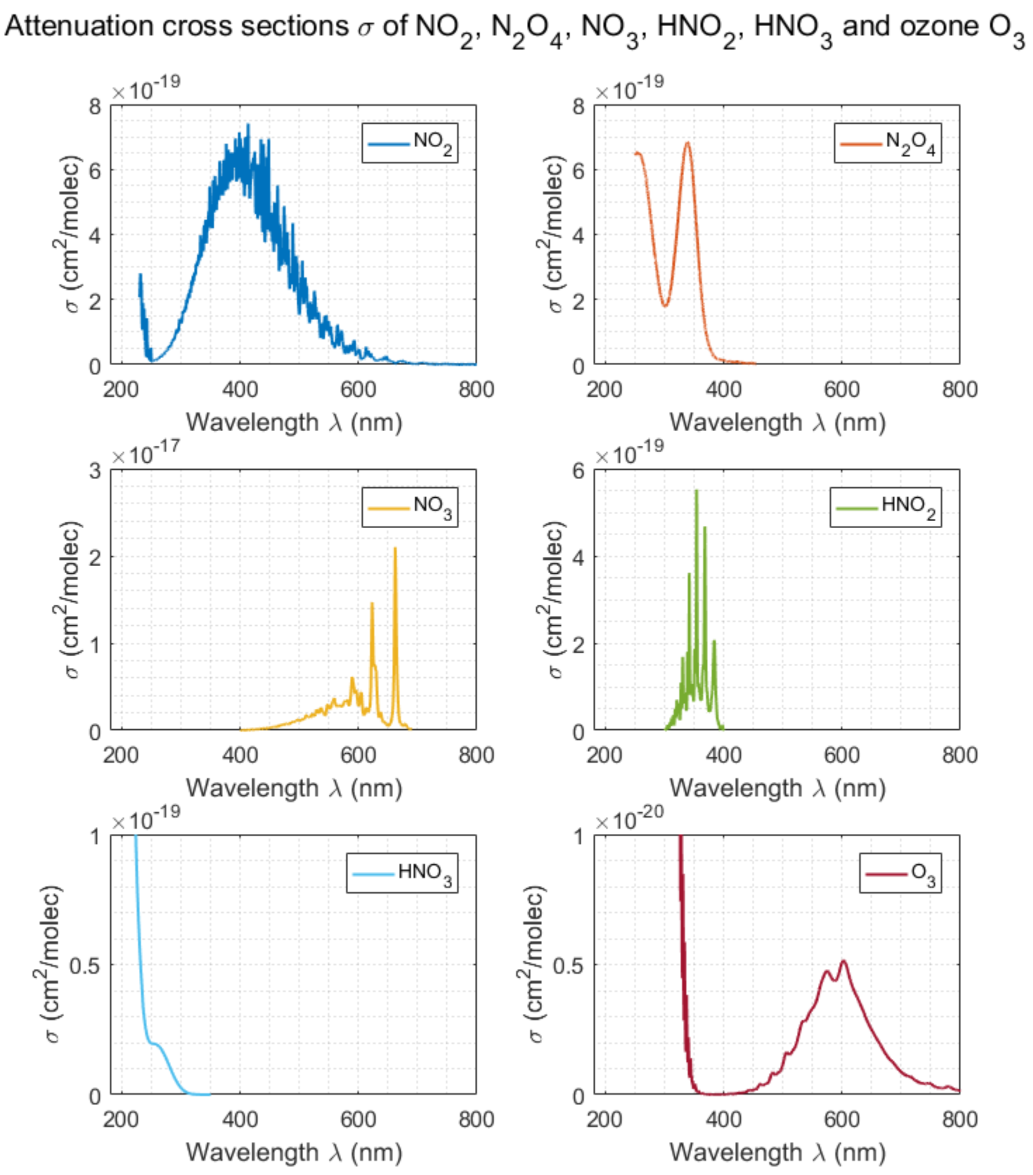


Figure 2.8: Attenuation cross sections of nitrogen dioxide NO_2, dinitrogen tetroxide N_2O_4, nitrate radical NO_3, nitrous acid HNO_2 and nitric acid HNO_3 as well as ozone O_3 [22]

The above attenuation cross sections σ are based on the pure gases and are given in area per molecule (cm^2/molec). To obtain the absorption spectrum in a mixture of gases, the attenuation coefficients μ for each gas i can be calculated if the volume fractions Φ_i or the number densities n_i are known. According to Equations (2.6) and (2.9) and the typical volume fractions, the different attenuation coefficients are determined and illustrated in Figure 2.9. The background attenuation of air is represented by the USA model of air from the HITRAN database [23].

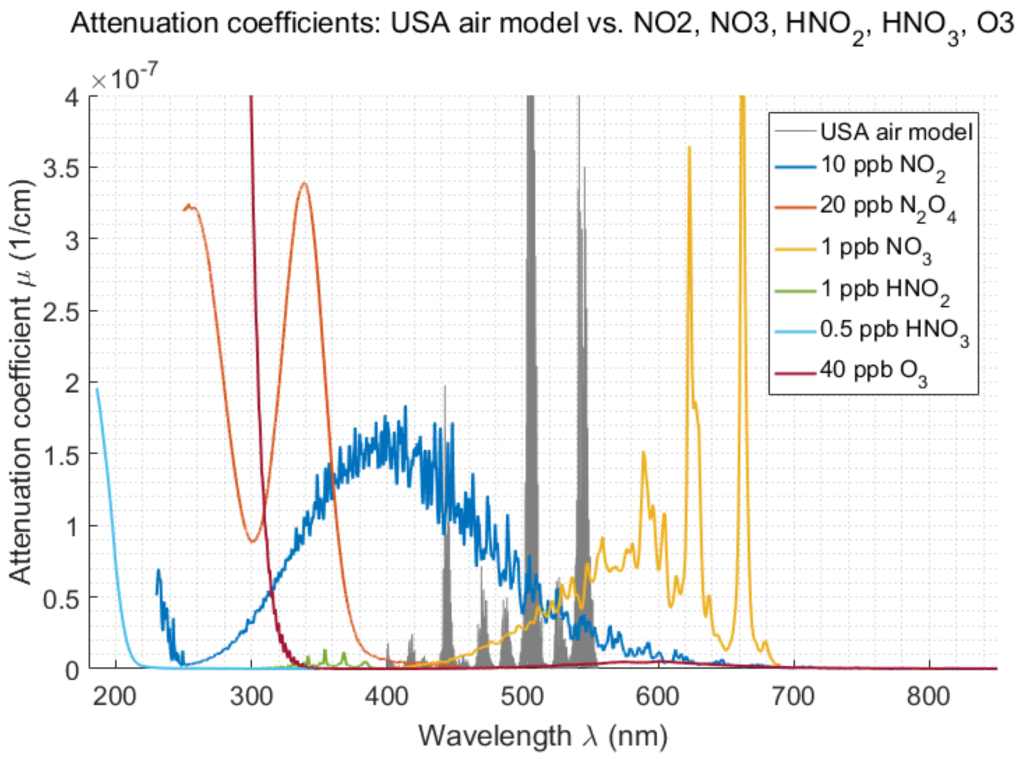


Figure 2.9: Attenuation coefficients of nitrogen dioxide NO_2, dinitrogen tetroxide N_2O_4, nitrate radical NO_3, nitrous acid HNO_2, nitric acid HNO_3 and ozone O_3 [22] with the USA model of air as background attenuation [23]

From the spectrum of attenuation coefficients the possible wavelengths for NO_2 measurements without interference of other gases can be observed. The most efficient ranges for excitation are therefore between 370 - 440 nm, 447 - 469 nm and 474 - 484 nm.

2.3.2 Photolysis

NO_2 is an atmospheric molecule, that is photochemical active in the UV and visible region of the spectrum. Its main absorption spectrum reaches from 250 - 600 nm and its photodissociation (photolysis) already occurs in the visible range. The photolysis of NO_2 is a chemical reaction induced by electromagnetic radiation energy $h\nu$ and generates nitric oxide NO and ozone O_3 in the presence of oxygen O_2 as per the chemical equation (2.20). It is furthermore the major source of O_3 in the troposphere.

$$\begin{gathered} NO_2 + h\nu \rightarrow O(^3P) + NO \\ O(^3P) + O_2 + M \rightarrow O_3 + M, \end{gathered} \tag{2.20}$$

where M is a 'third body' (e.g. molecule) to discard energy by collision [24].

The slower oxidation reactions of NO reduce the NO concentration by different mechanisms, for example:

$$\begin{gathered} NO + O_3 \rightarrow NO_2 + O_2, \\ 2NO + O_2 \rightarrow 2NO_2, \\ NO + O_2 \rightarrow NO_3, \quad NO_3 + NO \rightarrow 2NO_2, \\ NO + RO_2 \rightarrow RO + NO_2, \end{gathered} \tag{2.21}$$

where R refers to an alkyl or acyl radical [24–26].

The photodissociation threshold for NO_2 is reported at a wavelength of 397.9 nm [24] and corresponds to the required energy for dissociation. At this dissociation threshold the quantum yield of photolysis $\Phi(\lambda)$ is above 80 % and decreases rapidly for smaller radiation energies. Below the threshold NO_2 partially dissociates due to the contribution of internal energy (vibrational, rotational) [24, 27, 28]. The data for $\Phi(\lambda)$ at 298 K is out of Gardner's work [27]. This quantum yield is shown in Figure 2.10.

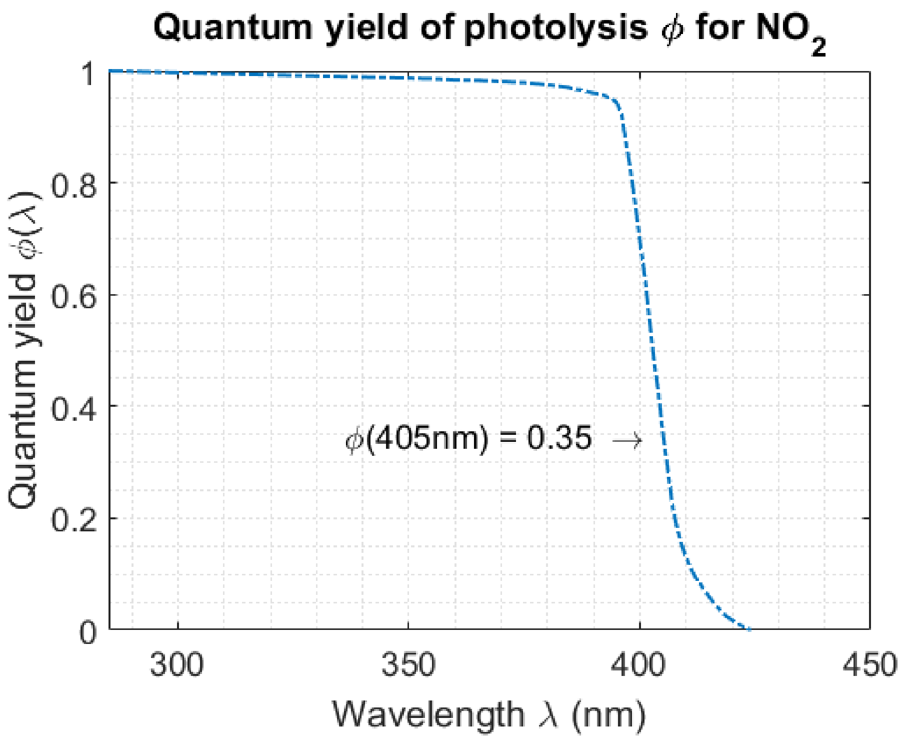


Figure 2.10: Best fit corrected data of quantum yield $\Phi(\lambda)$ of photolysis for NO_2 at 298 K, the marked data at 405 nm denotes the quantum yield for an available laser diode with that wavelength [27].

The photolysis of NO_2 for wavelength larger than 424 nm drops down to zero, for wavelengths smaller than 285 nm the quantum yield is a 100 % [27].

The effect of photolysis on photoacoustic spectroscopy has been described by Harshbarger and Robin. They compared the optical absorption spectrum of NO_2 with their measured opto-acoustic spectrum in the range from 7000 to 3000 Å (700-300 nm) in Figure 2.11. Both spectra look quite similar in the region, where photodissociation is zero ($\lambda > 4240$ A). With increasing quantum yield Φ of photolysis the photoacoustic signal declines. The cause is the loss of NO_2 molecules due to the photodissociation. The fact, that there is still a photoacoustic signal, can be explained by the rapid recombination of NO and O in combination with the pressure increase as a result of the generation of two molecules out of one nitrogen dioxide (NO_2 into NO and O) [28].

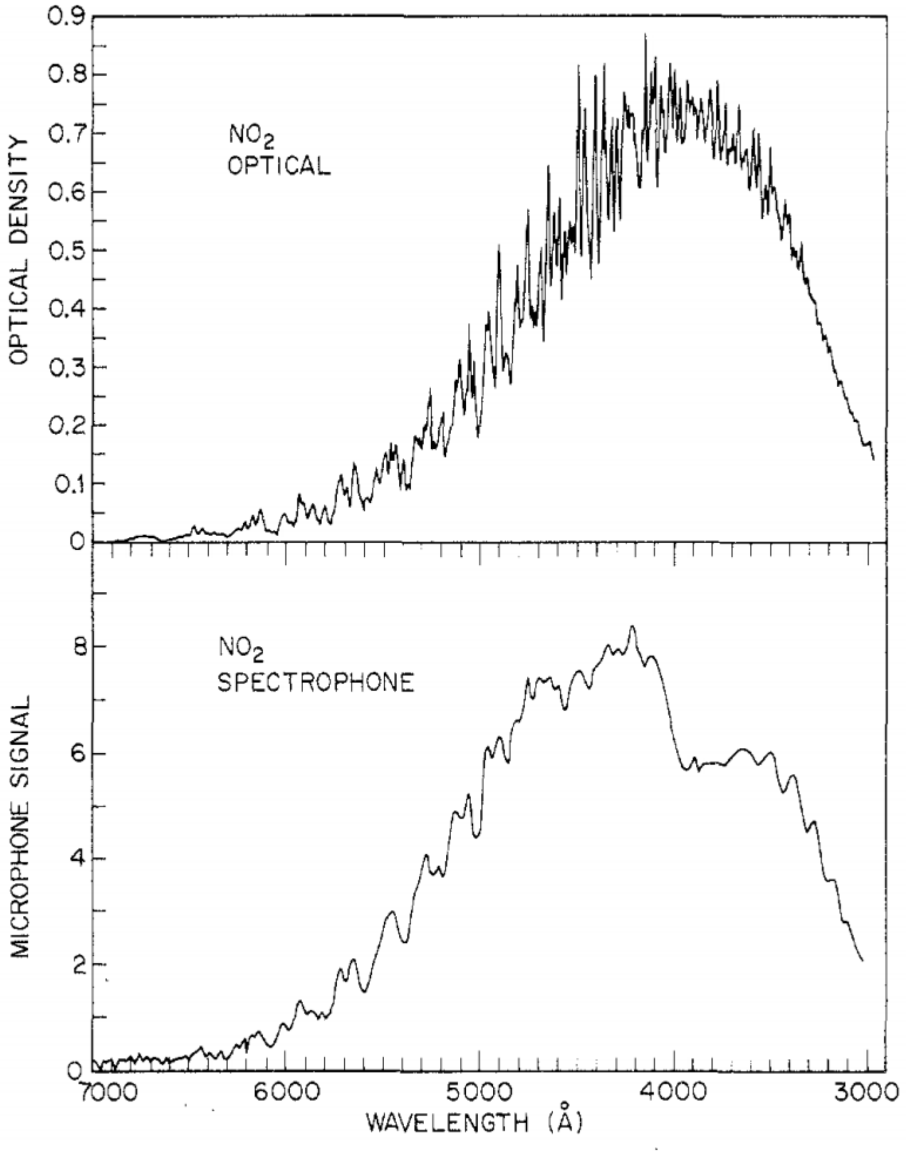


Figure 2.11: The comparison of optical absorption and photoacoustic (spectrophone) spectrum of NO_2 by Harshbarger and Robin shows a decrease of the microphone signal in the area of photodissociation [28].

2.3.3 Dimerisation

The concentration of NO_2 in air is strongly dependent on temperature and pressure. This is caused by the formation of its dimer N_2O_4. The chemical equilibrium of this dimerisation at a certain temperature and pressure is expressed in Equation (2.22). With increasing temperature T the reaction is shifted towards NO_2, with increasing pressure p towards N_2O_4.

$$N_2O_4 \underset{\leftarrow p}{\overset{T \rightarrow}{\rightleftharpoons}} 2NO_2 \tag{2.22}$$

For the quantitative description of the equilibrium the reaction Gibbs energy $\Delta_r G$ is utilized for the calculations. In the following the dissociation reaction of N_2O_4 will be observed with its reaction equation

$$N_2O_4 \rightarrow 2NO_2 \tag{2.23}$$

and their corresponding stoichiometric numbers v_J of the substances J with the values of $v_{N_2O_4} = -1$ and $v_{NO_2} = 2$. Considering the perfect gas equilibrium, $\Delta_r G$ is defined as

$$\begin{gathered}\Delta_r G = \Delta_r G^{\Theta} + RT \cdot lnQ, \\ \text{where}\end{gathered} \tag{2.24}$$

$$Q = \frac{\text{activities of products}}{\text{activities of reactants}} = \prod_J a_J^{v_J}, \tag{2.25}$$

with the standard reaction Gibbs energy $\Delta_r G^o$, the gas constant $R = 8.314\,459\,8\,\frac{\mathrm{J}}{\mathrm{mol\,K}}$, the reaction quotient Q and the substances' activities a_J [29]. Further $\Delta_r G^o$ can be calculated from

$$\Delta_r G^o = \sum_{Products} |v| \cdot \Delta_f G^o - \sum_{Reactants} |v| \cdot \Delta_f G^o = \sum_J v_J \cdot \Delta_f G^o, \tag{2.26}$$

where $\Delta_f G^o$ are the standard Gibbs energies of formation of the elements in their reference state [29].

At the stationary equilibrium state $\Delta_r G = 0$ and the activities a_J have their equilibrium values, so that the reaction quotient Q can be replaced by the equilibrium constant K, which can also be expressed in terms of the partial pressures for ideal gases: $a_J = p_J/p^o$ with the standard pressure $p^o = 1\,bar$. For the N_2O_4 reaction (Eq. (2.23)) K is given as

$$K = \frac{(a_{NO_2})^2}{a_{N_2O_4}} = \frac{\left(\frac{p_{NO_2}}{p^o}\right)^2}{\frac{p_{N_2O_4}}{p^o}} = \frac{(p_{NO_2})^2}{p_{N_2O_4} \cdot p^o}. \tag{2.27}$$

	standard Gibbs energy $\Delta_f G^o$ (kJ/mol)	standard enthalpy (kJ/mol)
NO_2	$+51.31$	$+33.18$
N_2O_4	$+97.89$	$+9.16$

Table 2.2: Standard Gibbs energy and enthalpy for NO_2 and N_2O_4 at standard temperature 298 K and pressure 1 bar [29]

By rewriting Equation (2.24) to solve for K at equilibrium $\Delta_r G = 0$ (note[4])

$$lnK = -\frac{\Delta_r G^o}{RT} \quad \rightarrow K = exp\left(-\frac{\Delta_r G^o}{RT} \right) \tag{2.28}$$

and calculating the standard reaction Gibbs energy $\Delta_r G^o$ according to Equation (2.26) using the values for $\Delta_f G^o$ of Table 2.2

$$\Delta_r G^o = 2 \cdot \Delta_f G^o(NO_2,\ g) - \Delta_f G^o(N_2O_4,\ g) = 4.73\ \frac{kJ}{mol}, \tag{2.29}$$

the equilibrium constant for the dissociation of $N_2O_4 \rightarrow NO_2$ at the standard temperature $T^o = 298\,\mathrm{K}$ and pressure $p^o = 1\,\mathrm{bar}$ is determined to $K = 0.1482\,\mathrm{kJ/mol}$.

The dissociation degree α is per definition the fraction of reactants that have decomposed of an initial amount n to the amount at equilibrium n_{eq}, $\alpha = (n - n_{eq})/n$. In order to determine this numerically, the equilibrium constant K has to be expressed in terms of α and then solved [29]. By drawing following table this can be related to each other.

$N_2O_4 \rightarrow 2\ NO_2$			
	N_2O_4	NO_2	total
initial amount	n	0	n
amount at equilibrium	$(1-\alpha)n$	$2\alpha n$	$(1+\alpha)n$
mole fraction, x_J	$\frac{1-\alpha}{1+\alpha}$	$\frac{2\alpha}{1+\alpha}$	1
partial pressure, $p_J = x_J \cdot p$	$\frac{(1-\alpha)p}{1+\alpha}$	$\frac{2\alpha p}{1+\alpha}$	p

Table 2.3: Table of equilibrium composition expressed in terms of dissociation degree α [29]

Hence, the equilibrium constant K is calculated through the partial pressures to

$$K = \frac{\left(\frac{p_{NO_2}}{p^o}\right)^2}{\frac{p_{N_2O_4}}{p^o}} = \frac{\left(2\alpha\frac{p}{p^o}\right)^2}{(1+\alpha)^2} \cdot \frac{1+\alpha}{(1-\alpha)\frac{p}{p^o}} = \frac{4\alpha^2}{1-\alpha^2} \cdot \frac{p}{p^o}. \tag{2.30}$$

4 $exp(x)$ in this case is the Euler potency e^x

Solving this equation for α, the dissociation degree can be determined for the equilibrium constant at standard conditions ($K = 0.1482\,\mathrm{kJ/mol}$).

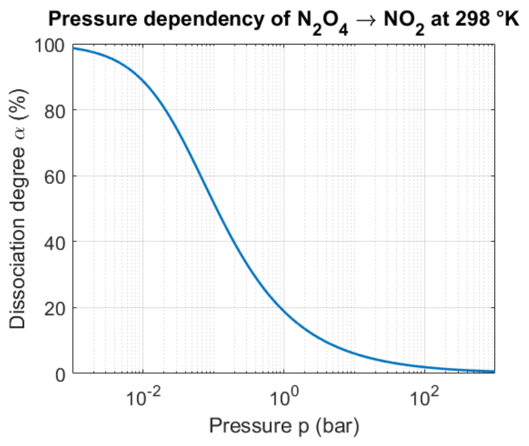
$$\alpha = \sqrt{\frac{K}{K + 4\frac{p}{p^o}}} = 0.189 = 18.9\,\% \tag{2.31}$$

This value corresponds to the percental amount of N_2O_4, that has dissociated. Considering that one molecule generates two NO_2, the mole fraction x_J is the better measure for the proportional share of the two gases

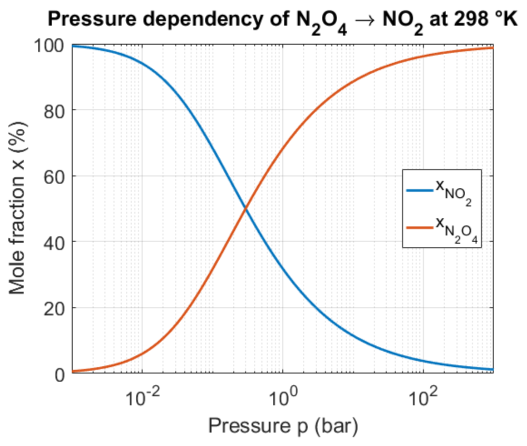
$$\begin{aligned} x_{N_2O_4} &= \frac{1-\alpha}{1+\alpha} = 0.682 = 68.2\,\%, \\ x_{NO_2} &= \frac{2\alpha}{1+\alpha} = 0.318 = 31.8\,\%. \end{aligned} \tag{2.32}$$

Equilibrium response to pressure

The standard reaction Gibbs energy $\Delta_r G^o$ is defined at a single standard pressure and is thus independent of pressure. So is the equilibrium constant K. For the perfect gas equilibrium only the partial pressures p_J can change, which are related to K as described in Equation (2.27). According to Le Chatelier's principle, the reaction of a system at equilibrium will adjust to minimize pressure, for instance, by reducing the number of particles in a gas [29]. Applying the independent K to Equation (2.31) for different pressures and using the corresponding α to calculate the mole fractions for N_2O_4 and NO_2 by Equations (2.32), the pressure dependency can be plotted as in Figure 2.12.



(a) Dissociation degree α of N_2O_4



(b) Mole fractions of NO_2 and N_2O_4

Figure 2.12: Pressure dependency of dissociation degree α **(a)** and mole fractions x_{NO_2}, $x_{N_2O_4}$ **(b)** from 1 mbar to 1 kbar at standard temperature 298 K at a logarithmic x-axis scale in bar

Equilibrium response to temperature

As mentioned before, $\Delta_r G^o$ and K are pressure independent, however temperature has a big influence. To derive this relation, the formula of Eq. (2.28) is differentiated with respect to T as follows [29]

$$\begin{aligned} lnK &= -\frac{1}{R} \cdot \frac{\Delta_r G^o}{T} \quad \Bigg| \cdot \frac{d}{dT}, \\ \frac{d(lnK)}{dT} &= -\frac{1}{R} \cdot \frac{d}{dT}\left(\frac{\Delta_r G^o}{T}\right). \end{aligned} \tag{2.33}$$

To develop the differentiation, Gibbs-Helmholtz equation is used in the form

$$\frac{d}{dT}\left(\frac{\Delta_r G^o}{T}\right) = -\frac{\Delta_r H^o}{T^2}, \tag{2.34}$$

where $\Delta_r H^o$ is the standard reaction enthalpy. By merging the two relations that gives the the van't Hoff equation

$$\begin{gathered} \frac{d(lnK)}{dT} = \frac{\Delta_r H^o}{RT^2} \\ \text{and its second form by the substitution } dT = -T^2 d(\tfrac{1}{T}) \\ \frac{d(lnK)}{d\frac{1}{T}} = -\frac{\Delta_r H^o}{R}. \end{gathered} \tag{2.35}$$

The integration of the second form provides the deviation of the equilibrium constant K_2 at a temperature T_2 to a reference K_1 at another temperature T_1 [29].

$$lnK_2 - lnK_1 = -\frac{1}{R}\int_{1/T_1}^{1/T_2} \Delta_r H^o d\tfrac{1}{T} \tag{2.36}$$

Unfortunately $\Delta_r H^o$ has a slight temperature dependency, but since there is often no other method available, the linearisation of $\Delta_r H^o$ is a reasonable practice. In most of the cases this temperature dependency is weak and the results therefore quite accurate [29]. Assuming the reaction enthalpy constant over temperature, the integral simplifies to [29]

$$lnK_2 - lnK_1 = -\frac{\Delta_r H^o}{R} \cdot \left(\frac{1}{T_2} - \frac{1}{T_1}\right) \tag{2.37}$$

and the solution for the K_2 then is

$$K_2 = exp\left(lnK_1 - \frac{\Delta_r H^o}{R} \cdot \left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right). \tag{2.38}$$

The standard reaction enthalpy $\Delta_r H^o$ is determined analogous to $\Delta_r G^o$ (Eq. (2.26)) with the standard enthalpies $\Delta_f H^o$ of the products and reactants.

For the dissociation of N_2O_4 into NO_2 this is calculated employing the values for $\Delta_f H^o$ out of Table 2.2.

$$\Delta_r H^o = \sum_J v_J \cdot \Delta_f H^o = 2 \cdot \Delta_f H^o(NO_2,\ g) - \Delta_f H^o(N_2O_4,\ g) = 57.2\ \frac{kJ}{mol} \tag{2.39}$$

As an example the equilibrium response is determined at a temperature of 323 K (around 50 °C). The temperature reference T_1 is at standard temperature of 298 K, wherefore the value of K_1 is already present. It follows then for K_2 by applying Equation (2.38) and the previous calculated value for $\Delta_r H^o$

$$K_2(323\,^\circ K) = \sqrt{lnK_1 - \frac{\Delta_r H^o d}{R} \cdot \left(\frac{1}{323\,^\circ K} - \frac{1}{298\,^\circ K}\right)} = 0.885. \tag{2.40}$$

For the dissociation degree α at a pressure of 1 bar that results in

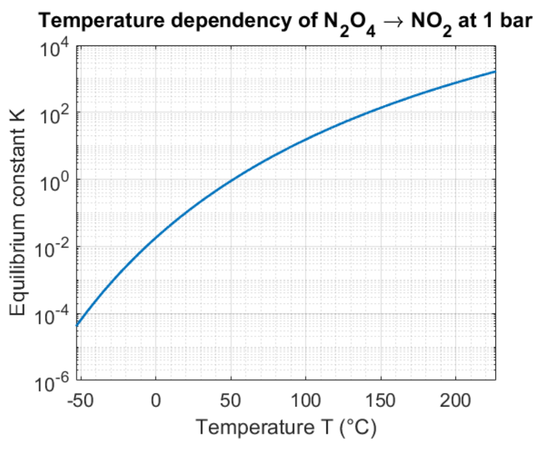
$$\alpha = \sqrt{\frac{K_2}{K_2 + 4\frac{p}{p^o}}} = 0.4256 = 42.56\,\% \tag{2.41}$$

and the two mole fractions consequently are

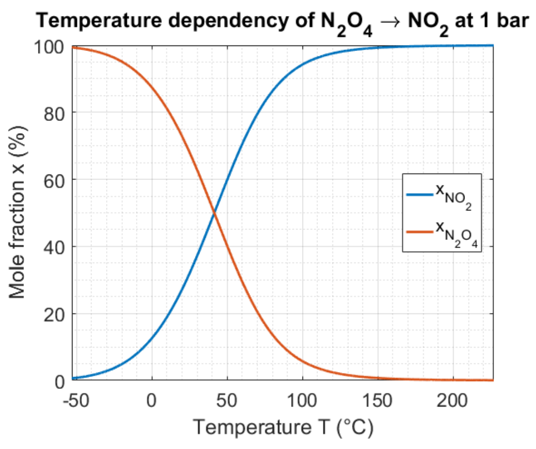
$$\begin{aligned} x_{N_2O_4} &= \frac{1-\alpha}{1+\alpha} = 0.4029 = 40.29\,\%, \\ x_{NO_2} &= \frac{2\alpha}{1+\alpha} = 0.5971 = 59.71\,\%. \end{aligned} \tag{2.42}$$

That corresponds to almost twice as much NO_2 at a temperature increase of only 25 K/°C.

The dependency of the equilibrium constant K **(a)** and the mole fractions x_{NO_2}, $x_{N_2O_4}$ on temperature using the above demonstrated method is illustrated in Figure 2.13.



(a) Equilibrium constant K for the reaction $N_2O_4 \rightarrow NO_2$ at a logarithmic y-axis scale



(b) Mole fractions of NO_2 and N_2O_4

Figure 2.13: Temperature dependency of the equilibrium constant K **(a)** and the mole fractions x_{NO_2}, $x_{N_2O_4}$ **(b)** from −50 °C to 250 °C at standard pressure of 1 bar

2.4 Conclusion

In general the detection of NO_2 gas based on the QEPAS method is possible. There is a restriction in the wavelength range for the laser excitation caused, on the one hand, by the absorption spectrum of NO_2 and other present gases in air as well as the background attenuation of air (cf. Figure 2.8). On the other hand, this is limited by the photolysis of NO_2, that decomposes the molecule for wavelengths $\lambda < 424\,\mathrm{nm}$, which might be corrected for less than 50 % of photolysis quantum yield, but would require further investigations. Since there is a limited selection of available laser diodes in the effective wavelength range, the decision for a 450 nm laser (OSRAM PL 450B) was made. The spectral relative emission spectrum for this laser diode is therefore inserted in the attenuation spectrum in Figure 2.14 to illustrate the small impact of other gases.

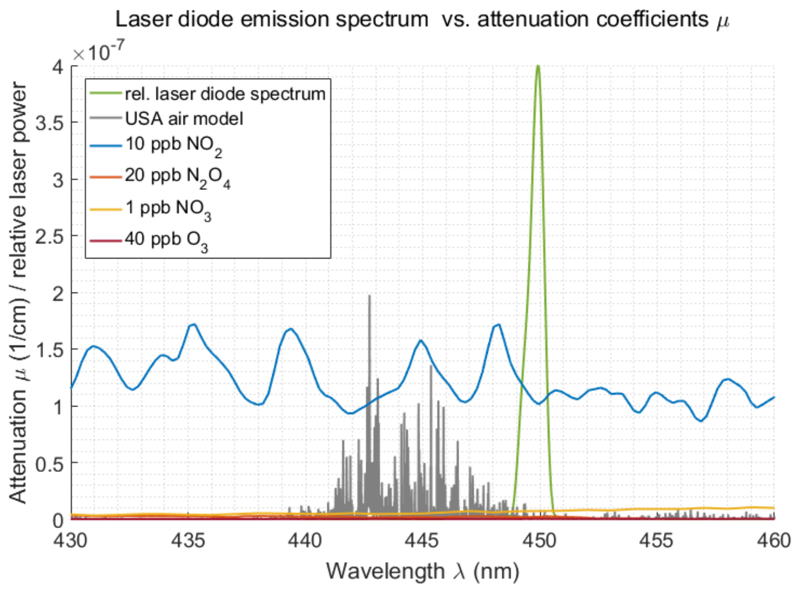


Figure 2.14: Relative spectral emission of an 450 nm laser diode (OSRAM PL 450B) compared to the relevant gas attenuation coefficients for common concentrations

The difficulty of the pressure and temperature dependency of NO_2 concentration due to dimerisation can also be handled in two ways. Either the concentration measurement, meaning the measurement setup and the inlet air, is temperature-stabilized or the correction is done mathematically by use of the mentioned equilibrium equations, whereby the decreasing concentration can get problematic for the detection limit. A combination of heating and mathematical correction is conceivable as well. The influence of pressure for ambient measurements should be insignificantly small compared to the temperature dependency, because of the nearly constant atmospheric pressure.

CHAPTER 3

Photoacoustic Setup

The photoacoustic setup has been established by Philipp Breitegger (Institute of Electronic Sensor Systems, TU Graz). The setup applies quartz-enhanced photoacoustic spectroscopy (QEPAS) in the visible range of the NO_2 absorption for a precise measurement of NO_2 concentrations. For the operation of the system expensive equipment is required, which consists of a National Instruments PXI measurement unit containing a signal generator and an Analog-to-Digital Converter (ADC) as well as a laser driver.

The main focus of this thesis is to replace these components by custom circuits. The developed solution includes the laser driver with built-in frequency generation and the dual-phase lock-in amplifier (LIA) for weak signal detection. Because of the narrow resonance behaviour of the quartz tuning fork (QTF), the frequency generation has to be very precise, which can be realized by direct digital synthesizers (DDS). For the dual-phase LIA it is additionally necessary to have a $90°$ phase-shifted reference signal. In this realization this is achieved by using a second DDS, which is synchronized to the first one providing a sine and cosine reference signal. The two circuits are controlled by a microcontroller communicating to the IC components via SPI or I2C interface. The COM port allows to send measurement commands from the PC (MATLAB script) to the microcontroller and receive the measured data.

Figure 3.1 gives an overview of the QEPAS setup and the peripheral equipment. The measurement chamber encapsulates the measurement from ambient air and makes it possible to control the NO_2 concentration through a gas in- and outlet. The 450 nm laser diode is focused between the prongs of the QTF through optical lenses and the weak current signal from the QTF is pre-amplified directly at the chamber.

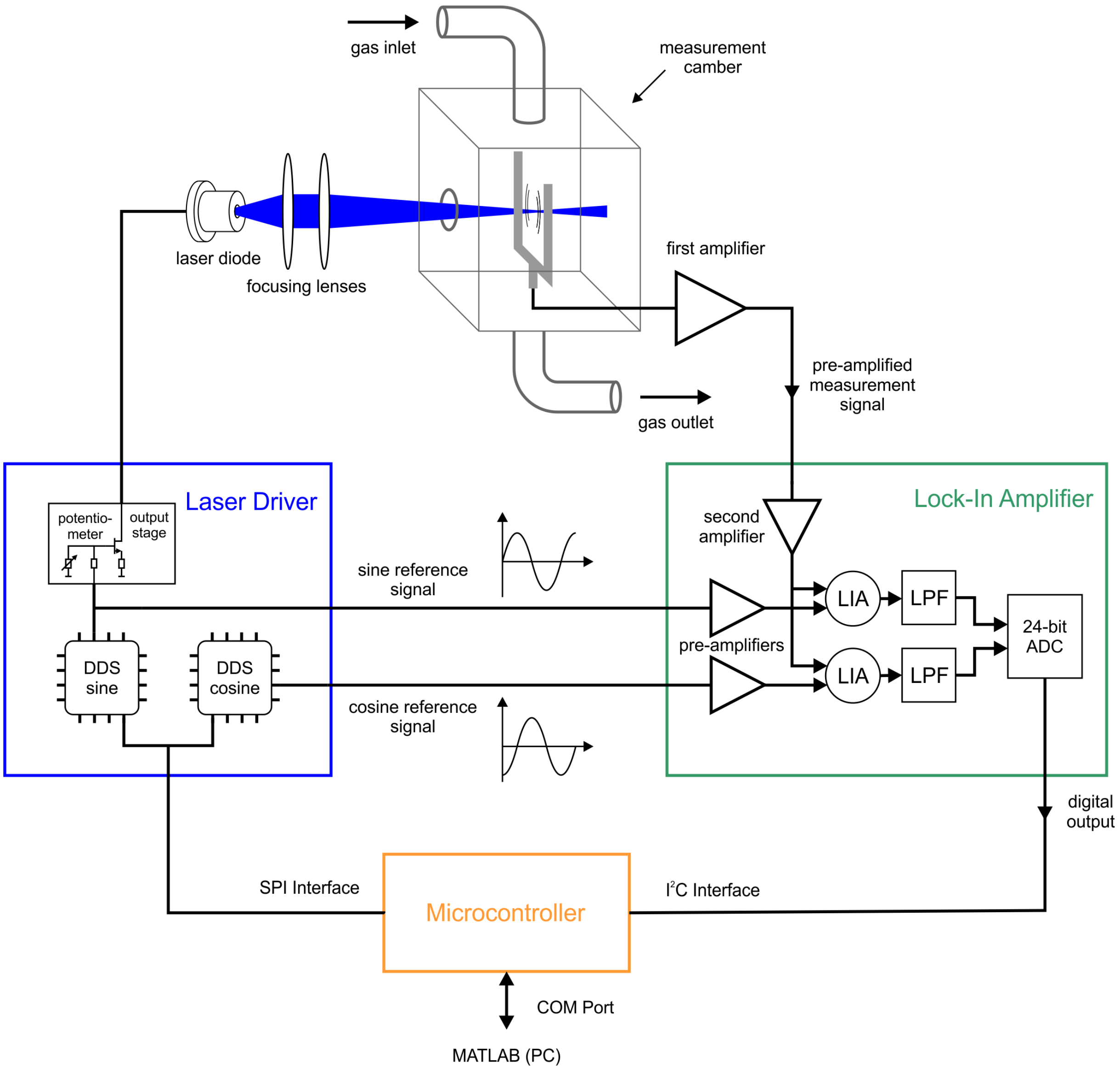


Figure 3.1: Functional principle of the measurement setup and overview of the circuit components and their functionality.

The development of the laser driver (blue box), the lock-in amplifier (green box) and the implementation of the microcontroller (yellow box) are described in the following chapters.

CHAPTER 4

Laser Driver

This chapter is dedicated to the circuit design process of the laser driver. The introduction section describes the main idea based on the targeted specifications, the working principle of the frequency generation (sine and cosine) for the laser modulation. This block as well as the current output stage and the current control circuit using digital potentiometers will than further be explained regarding their electronic schematic in the circuit design section. Afterwards some simulation results are presented and finally the functionality is investigated and important modifications are pointed out.

4.1 Introduction

This laser driver is designed to supply two laser diodes with a modulated current signal with a precise frequency resolution and controllable current amplitude. Additionally, there are two DC current outputs, which could be used as voice coil drivers for an optical lens adjustment in vertical and horizontal direction. Such optics are found as part of Blue-Ray laser modules, which could be used as cheap focusing optics for the photoacoustic setup in the future version. The circuit includes its own frequency generation using two direct digital synthesizers (DDS) and can be supplied by a single supply voltage of 12 V, that is regulated to the requires voltage levels for the analog and digital circuitry by low-dropout (LDO) regulators.

4.1.1 Concept and Specification

The current driving output stage is built with a precision current sink operating an NMOS transistor. The voltage at a shunt resistor is the feedback for the operational amplifier (OPA). Its input is the frequency signal generated by the DDS, the amplitude of this signal defines the current amplitude. To vary the voltage amplitude a voltage divider is arranged using a fixed resistance and a digital controllable potentiometer/rheostat. The specifications for the laser driver circuit are listed below.

Functional specifications:

- **Laser current output:** 0 mA to 100 mA for square wave output
- **Laser diode forward voltage:** 4 V up to 7 V
- **AC and DC operation should be possible**
- **Duty cycle:** 50 %
- **Frequency range:** 31 kHz to 34 kHz
- **Minimum frequency resolution:** ≤0.5 Hz
- **Required reference signals:** sine and 90° phase-shifted signal (cosine)

4.1.2 Synchronous Sine and Cosine Generation

For the frequency generation for this particular application it is important to select a DDS, that is either capable to produce the sine as well as the cosine wave, or to use a combination of two DDSs, that are able to be synchronized to each other. Furthermore, the IC should be programmable via an SPI or I2C interface and operate in the required frequency range with the required resolution (<0.5 Hz).

The best solution was found to be the AD9834 from Analog Devices in combination with an 8 MHz oscillator as reference clock, which provides a resolution of ∼30 mHz at a sampling rate of 250 $\frac{\text{samples}}{\text{period}}$ based on a 32 kHz output signal and a ∼0.09° phase resolution. The single output DDS has no internal synchronization function to a second one, though both can be synchronized by the microcontroller later on forcing a simultaneous counter register reset. Additionally this IC offers a square wave output parallel to the sinusoidal signal, which is used for the laser modulation.

4.2 Circuit Design

This section presents the circuit design of the important blocks of the laser driver. For the schematic and PCB design the program KiCad was used and, if the simulation models were available, some simulations were done in LTspice. Otherwise the datasheets and application notes of the IC components were used as reference for the electrical design. The full schematic and PCB layout is provided in the appendix A.

4.2.1 Direct Digital Synthesizer

The AD9834 is a low power, high performance DDS for sine wave generation with on-board comparator, that allows square wave output. It supports 3-wire SPI interface and offers two 28-bit frequency and two 12-bit phase, programmable registers. In this application the device in combination with an 8 MHz external oscillator is employed to generate a high-frequency-resolution sine and square wave signal in the range between 31 kHz to 34 kHz. In Figure 4.1 the circuit schematic based on the AD9834 datasheet [30] is shown for one of the two identical DDS designs. The power supply **Vdd_IC** is 5 V, the sine signal is available at the **sin** pin, the square wave at the **pulse** pin and the pins **SCLK** (SPI clock), **SDATA** (SPI data), **CS_DDS_I** (chip select) and **DDS_reset** (reset) build the SPI interface. Its detailed description follows below.

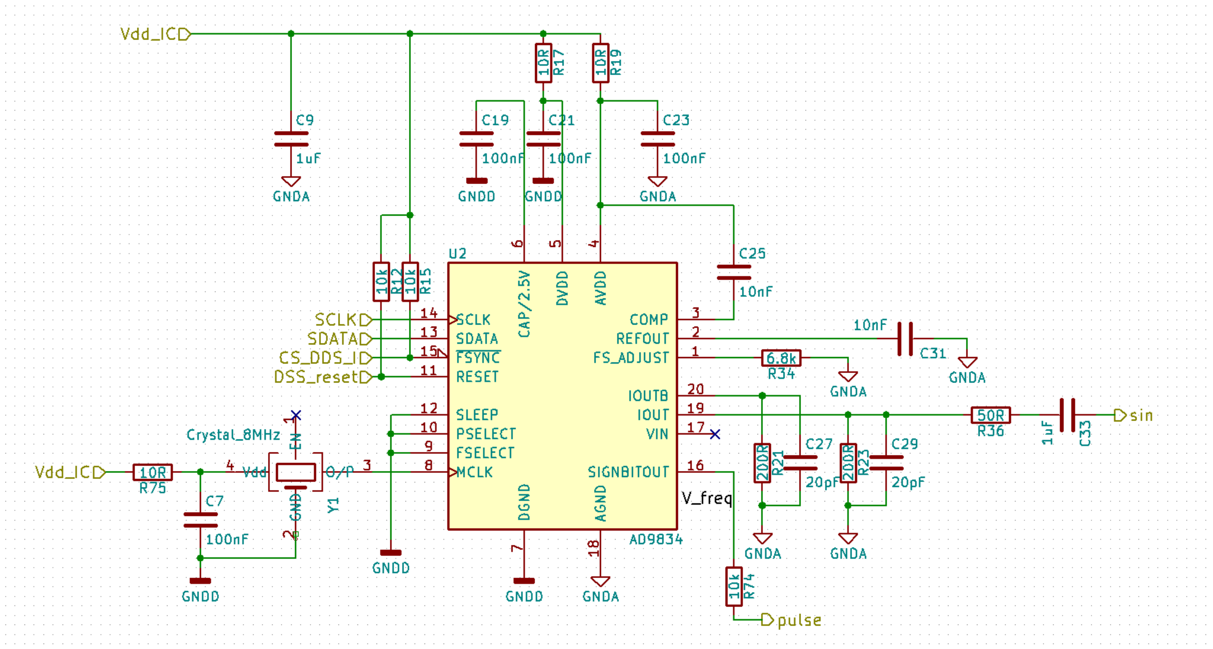


Figure 4.1: Schematic of sine and square wave generation circuit using the AD9834 DDS with an 8 MHz oscillator

Power Supply

The 5 V power supply **Vdd_IC** is decoupled from the rest of the circuit by a 1 µF capacitor **C9** before supplying the DDS and oscillator. The positive voltage supplies of **DVDD**, **AVDD** and **Vdd** of the oscillator are additionally decoupled from **Vdd_IC** by a current limiting resistance of 10 Ω and a 100 nF decoupling capacitor (e.g. **R17**, **C21**). For the on-board 2.5 V regulator of the DDS the pin **CAP/2.5V** requires to be decoupled to **GNDD** with a 100 µF capacitance (**C19**), when **DVDD** exceeds 2.5 V (otherwise connect to **DVDD**). Another decoupling for the DAC (Digital-to-Analog Converter) bias voltage is advised at the **COMP** pin and added by the 10 µF capacitor **C25** [30].

Analog Signals

Since the reference output **REFOUT** is unused this pin is kept stable through a 10 uF capacitance **C31** to ground. The **FS_ADJUST** pin controls the full-scale adjustment of the current output at the **IOUT** pin by applying a voltage value to **FS_ADJUST**. A resistor to ground (**R34**) with the typical value of 6.8 Ω sets the current output to 3.12 mA at full-scale. This produces an output voltage of $V_{max} = R \cdot I = 624\,\text{mV}$ over the 200 Ω resistance **R23**. The typical minimum output current is 0.16 mA and consequently the output voltage results to $V_{min} = 32\,\text{mV}$, which gives a peak-to-peak amplitude of approximaly $V_{pp} = 600\,\text{mV}$ [31]. For clock feedthrough prevention a 20 pF capacitor **C29** parallel to **R23** is recommended. The output **IOUTB** delivers the complementary sine wave for differential operation, which is not used, and therefore should have the same load as **IOUT** (**R21**, **C27**) [30]. A 50 Ω resistor **R36** and a 1 uF AC-coupling capacitor **C33** are inserted in front of the **sin** reference output. This signal is, besides the cosine reference of the second DDS output, one of the reference signal inputs for the lock-in amplifier circuit, where these signals will be properly amplified for the application (cf. Section 5.2.1).

Digital Interface and Control

The DDS master clock **MCLK** is fed by the 8 MHz quartz oscillator. Its enable pin **EN** is active for tristate/open or Vdd. The control pins **FSELECT** and **PSELECT** select which of the frequency/phase registers are used for signal generation. Since this can be control by a bit in the control register, they are disabled by connecting them to ground. The same is done for the active low **SLEEP** pin. As SPI interface the pins **SCLK**, **SDATA**, **CS_DDS_I** ($\overline{\text{FSYNC}}$) are of use, where an active low on **CS_DDS_I** activates the communication. The synchronization of both DDS is achieved with the **DDS_reset** pin, which sets the internal counter registers to zero, and is connected to the second DDS as well. To define a stable voltage level on the pins **CS_DDS_I** and **DDS_reset** for normal operation, they are connected to **Vdd_IC** through the pull-up resistors **R12** and **R15** (in case of power-on, absence or reboot of the microcontroller). The 3-wire SPI and the **DDS_reset** are directly connected to the microcontroller.

4.2.2 Digital Potentiometers

For the voltage divider network the digital potentiometer MCP4542-103E is used. This is a dual 10 kΩ resistor network in rheostat configuration, which means that it has two digital controllable resistors between the pins **PW0** and **PB0** and between **PW1**, **PB1**. The communication interface is I^{2}C and its 7-bit address can be configured by the digital pins **HVC/A0** and **A1**. Because of its volatile RAM memory type the register values are loaded to default values and the wiper[5] position is set to mid-scale on power-on [32]. In AC operation the modulation signal **V_freq** (5 V amplitude) from the DDS is applied to the voltage divider consisting of the 90 kΩ resistor (**R49**, **R51**) and the rheostat resistance of max. 10 kΩ. The output amplitude is consequently between 0 V and 0.5 V. Figure 4.2 shows the schematic of the voltage divider, which is further explained afterwards.

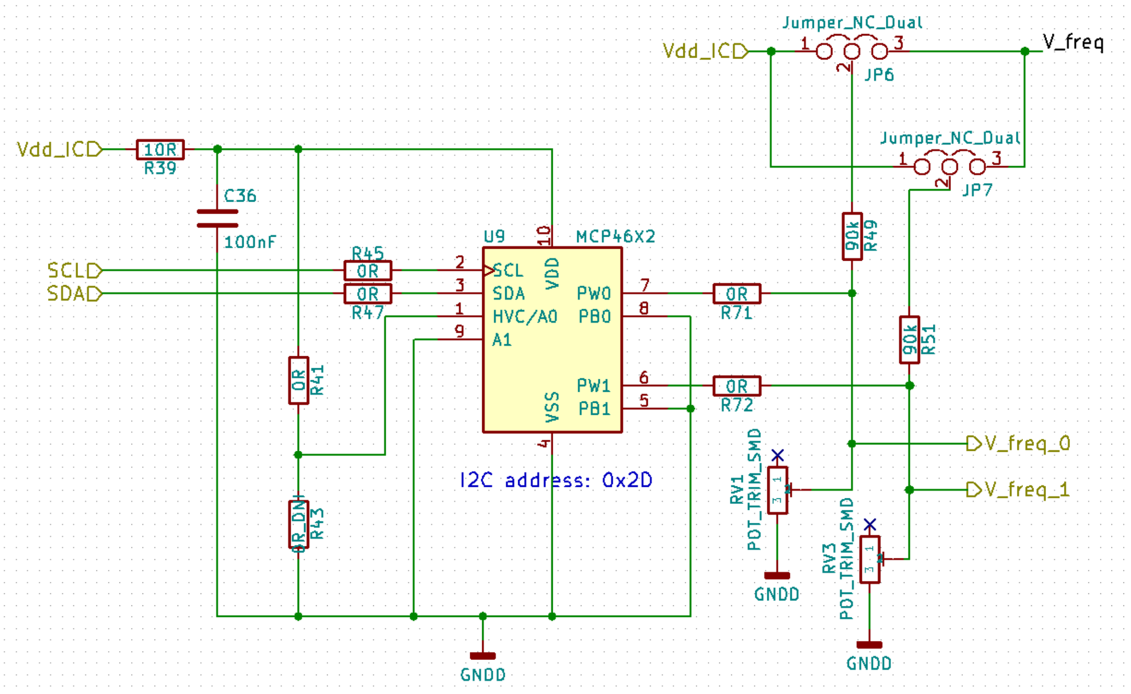


Figure 4.2: Schematic of the digital potentiometer voltage divider circuit for the modulation current amplitude control

[5] The wiper in this case is the digital switch, that defines the resistance value.

The MCP4542 can operate at the 5 V **Vdd_IC** voltage, that is decoupled with **R39** (10 Ω) and **C36** (100 nF). By the use of the 0 Ω **R45** and **R47** the I^{2}C interface (**SCL**, **SDA**) can be disconnected from the device for debugging purpose. The I^{2}C address is set to 0x2D by connecting the address pin **HVC/A0** to **Vdd_IC** using **R41**, omitting **R43** (DNI stands for: do not insert) and **A1** to ground **GNDD**. For testing of the circuit without programming the MCP4542, the digital potentiometer can be disconnected from the voltage divider by desoldering **R71**, **R72** and inserting the mechanical 10 kΩ potentiometers **RV1** and **RV3**. Switching between AC and DC operation mode is achieved by the jumper pin **JP6** and **JP7**. The two outputs **V_freq_0**, **V_freq_1** of the voltage divider are directly connected to the corresponding laser output stages. For the voice coil output stages the digital potentiometer circuit is the same except that the voltage divider is directly connected to **Vdd_IC** as they only need DC operation.

4.2.3 Precision Current Sink

The laser driver is realized by the use of a precision current sink. The schematic is given in Figure 4.3 and is basically the same for all four output stages, the two laser outputs and the two voice coil outputs. This is built with a low noise 88 MHz, single supply operational amplifier (OPA) LMV793 [33] and the fast switching N-channel MOSFET transistor DMG2302UK [34]. The laser diode is connected between **LD_anode** and **LD_cathode** and is supplied by 9 V from the **+9V_Driver** net, generated by an LDO regulator. This voltage is stabilized through the 10 µF capacitor **C5** and current limited by the 20 Ω resistor **R8** to 100 mA at maximum input **V_set** of 500 mV. Additionally, the Zener diode **D1** with a Zener breakdown voltage of 6.8 V is a voltage protection for the laser diode. The OPA regulates the voltage at the gate of the MOSFET transistor through the feedback loop including the resistors **R32**, **R6**, **R7** and the capacitance **C4** in order to match the input voltage **V_set** to the voltage at the 5 Ω source resistance **R9**. The current I_{DS} through the transistor and **R9** is then proportional to the input voltage $I_{DS} = \frac{V\_set}{5\Omega}$. For an input range from 0 V to 500 mV this results into a current between 0 mA and 100 mA. The stability of the current sink is improved by the 1 nF capacitor **C45** at the drain of the MOSFET and in addition the input voltage **I_set** is buffered in front of the amplifier using **C2** and **R4**. As in the circuits before the 5 V supply voltage (**+5V_Driver**) is decoupled by a resistance **R5** and capacitance **C3**.

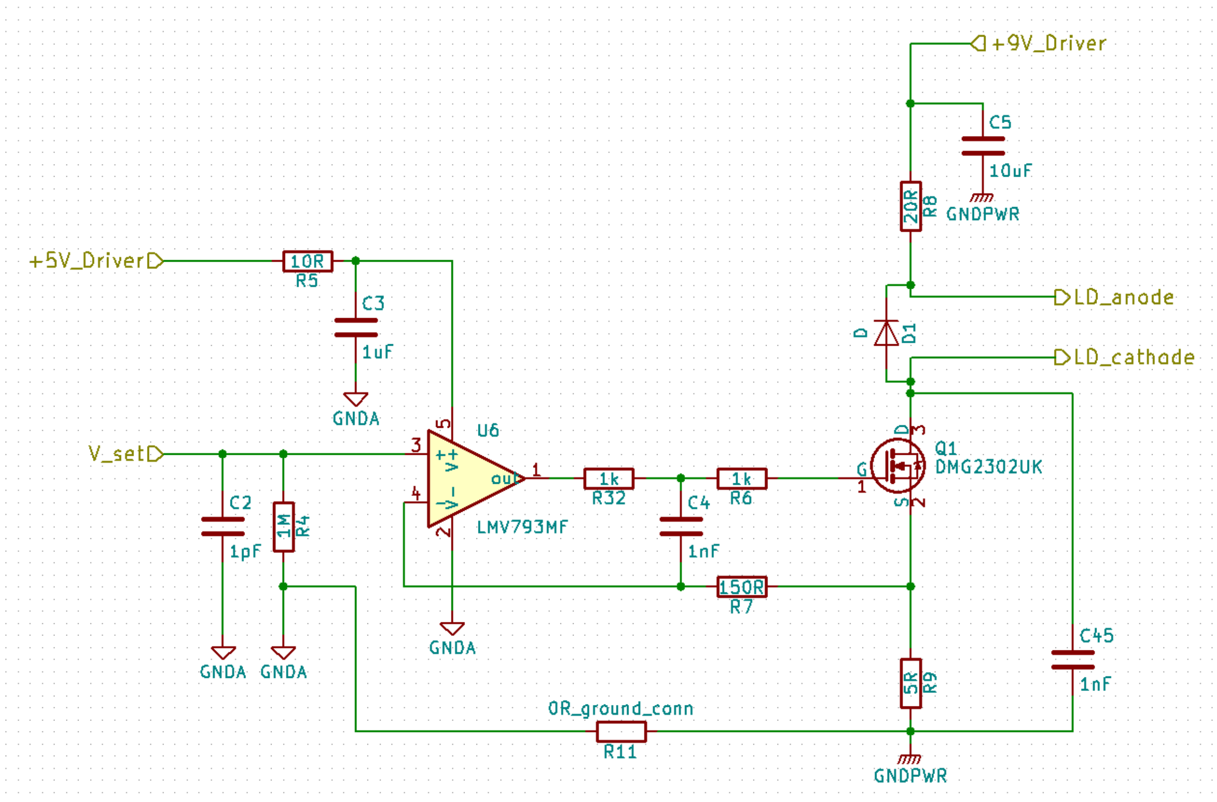


Figure 4.3: Schematic of the precision current output sink for driving the laser current

4.2.4 LDO Regulators and Ground Plane Concept

As mentioned before the 12 V power supply for the laser driver is converted to the required voltage levels for the different circuit parts by low-dropout (LDO) regulators. For the digital part, the supply voltage is 5 V. The analog amplifiers for the current sink are also operated at 5 V, generated by an additional LDO regulator, and the laser current is driven by the voltage of a 9 V regulator. For both 5 V voltages the regulator LP2950-5.0V [35] is used and the 9 V supply is generated by the LM2940-9.0V [36]. For the input and output capacitors the recommended values were applied. In Figure 4.4 the corresponding schematic is shown.

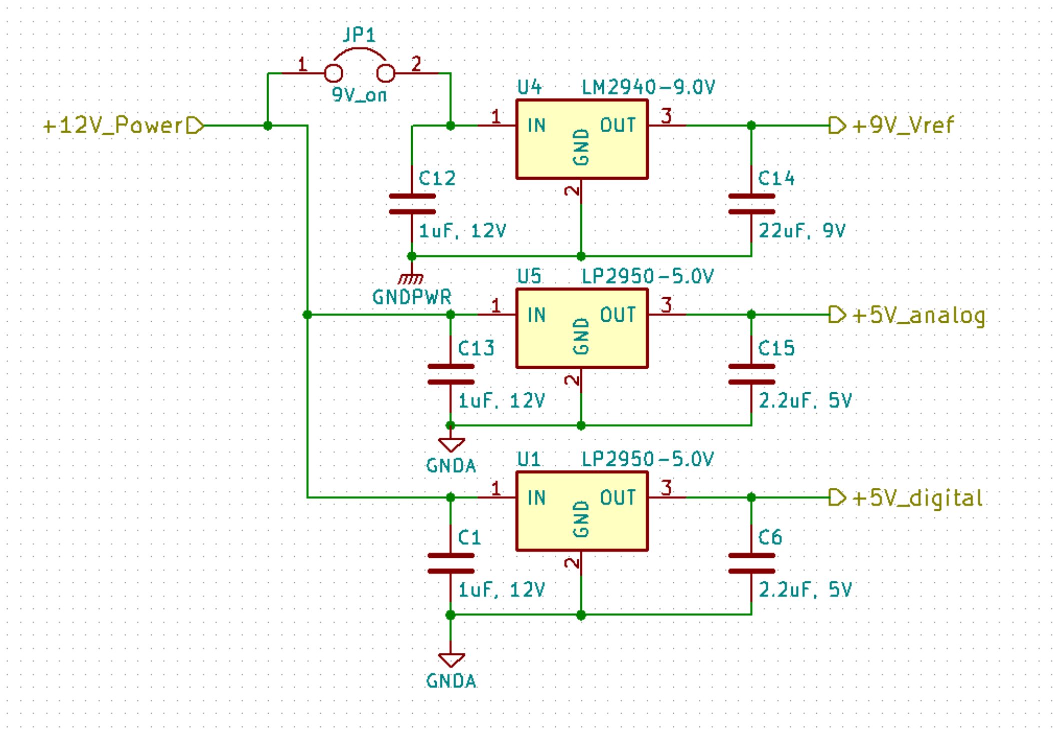


Figure 4.4: Schematic for the three low-dropout (LDO) regulators for the 5 V digital voltage supply **+5V_digital**, the analog **+5V_analog** and the laser current supply **+9V_Vref**

To minimize ground problems and improve performance a ground plane concept is used in the PCB layout. The three parts of digital, analog and laser current (power) circuit are split and have their own ground plane, which is connected at a single point of lowest potential, directly at power connector. Further, each of the circuit parts has a separate LDO regulator in order to have a stable voltage supply and not to interfere each other. This concept is illustrated in Figure 4.5.

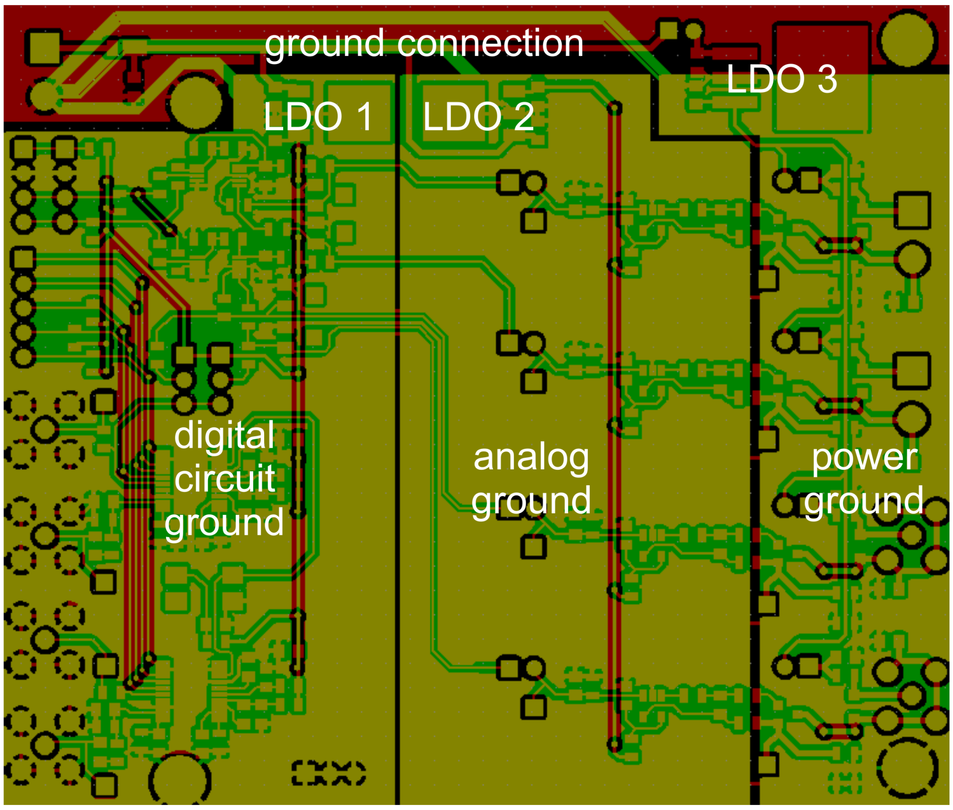


Figure 4.5: Ground plane concept: The digital, analog and power circuit have their own LDO regulator and a separate ground plane (top & bottom), which is connected at the supply ground.

4.3 Simulations

The simulations of the precision current sink are based on the schematic of Figure 4.3 and are performed in LTspice. In Figure 4.6 the behaviour of the laser diode current **I(D1)** is compared to the pulsed input voltage **V(v_in)** with 500 mV amplitude and a frequency of ∼33 kHz in a transient simulation. The output current is a nearly perfect square wave with an amplitude of 100 mV and a fast rise and fall time of less than 0.3 µs.

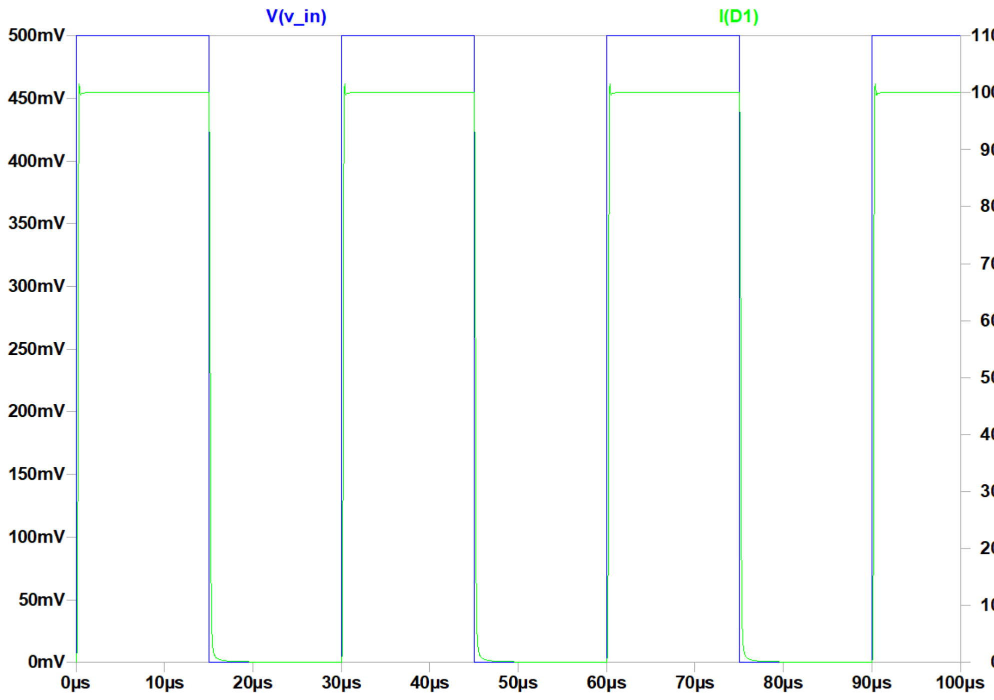


Figure 4.6: Simulation result: Behaviour of the laser current I(D1) of the precision current sink to a 500 mV pulsed voltage input V(v_in) with ∼33 kHz

Figure 4.7 shows the response of the laser current **I(D1)** to a amplitude increase of the input voltage **V_in** from 0 V to 0.5 V in steps of 0.1 V.

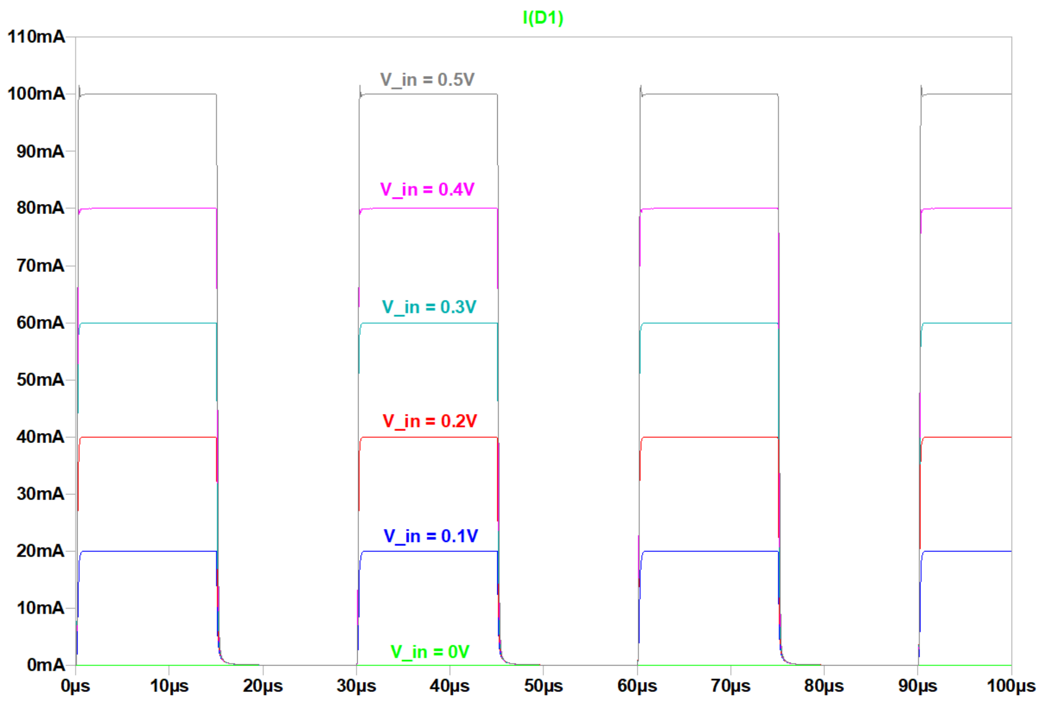


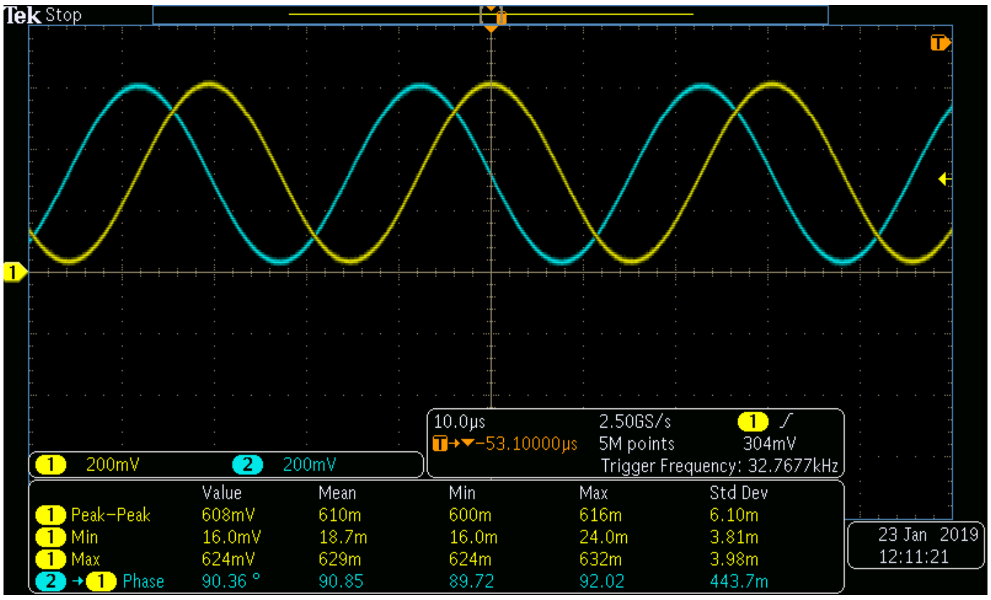
Figure 4.7: Simulation result: Laser current I(D1) behaviour for an increasing pulsed input voltage V_in from 0 V to 0.5 V in steps of 0.1 V.

4.4 Evaluation and Verification Measurements

This section covers the evaluation of the three main circuit parts of frequency signal generation using the DDS, digital potentiometer voltage division and the current sink. The configuration of the integrated devices ICs (DDS and digital potentiometers) is discussed in Chapter 6. At the end the most important modifications of the laser driver are noted.

4.4.1 DDS Signal Generation

The precise frequency generation of the sine and cosine signals is essential in order to match the modulation frequency to the resonance frequency of the QTF. As mentioned before, the calculated frequency resolution of the DDS is 30 mHz and the phase can be set in steps of 0.09°. Both signals at the 32.768 kHz resonance frequency are shown in the oscilloscope captures of Figure 4.8. To demonstrate the accuracy of the step size, a second capture at the resonance frequency plus 10 Hz is added in Figure 4.9. The square wave DDS output is discussed in Section 4.4.2, where this signal is applied to the voltage divider.



1 200mV 2 200mV

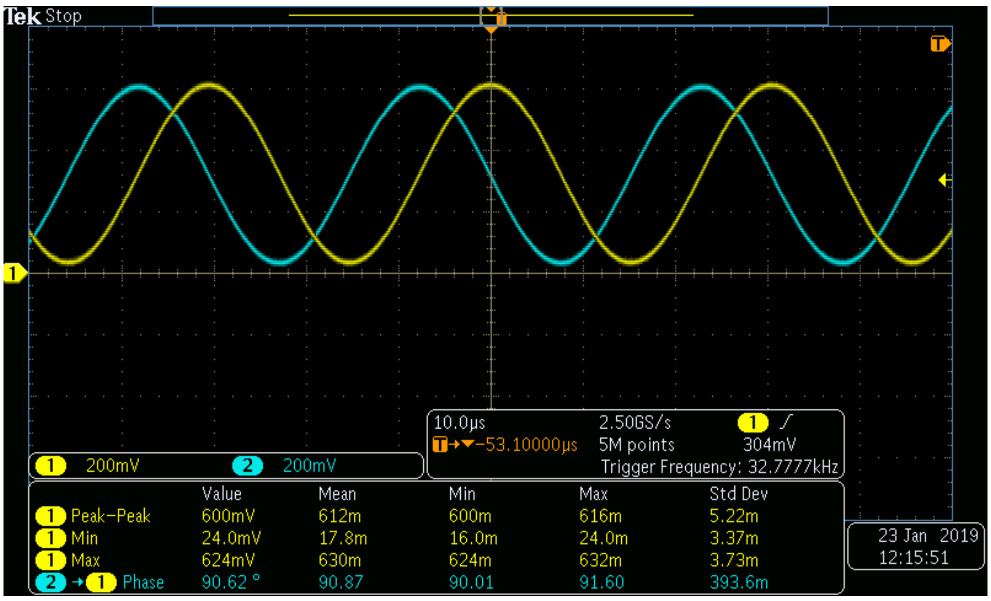
10.0µs 2.50GS/s 1 ⁄
T→▼−53.10000µs 5M points 304mV
Trigger Frequency: 32.7677kHz

	Value	Mean	Min	Max	Std Dev
1 Peak−Peak	608mV	610m	600m	616m	6.10m
1 Min	16.0mV	18.7m	16.0m	24.0m	3.81m
1 Max	624mV	629m	624m	632m	3.98m
2 → 1 Phase	90.36 °	90.85	89.72	92.02	443.7m

23 Jan 2019
12:11:21

Figure 4.8: Oscilloscope capture: Generated sine (yellow) and cosine wave (blue) at the set frequency of 32.768 kHz with peak-to-peak values of around 600 mV

The measured frequency (see Trigger Frequency in Figure 4.8) is 32.7677 kHz and hence −0.3 Hz off to the set frequency. Either this is caused by the limited oscilloscope accuracy or the deviation of the 8 MHz quartz oscillator master clock frequency. This is, however, irrelevant for our application, as the resonance frequency is determined with the same DDS that produces the modulation signal. As both use the same timebase. The measured phase of the cosine signal to the sine is 90.85° (weak phase accuracy of oscilloscope, better accuracy supposed). Investigating the amplitudes of both signals, the peak-peak values of around 600 mV (min. 16 mV, max. 624 mV) are as expected from the datasheet. The increase in frequency of 10 Hz (Figure 4.9) produces a measured frequency of 32.7777 kHz with the same −0.3 Hz offset (Δf =10 Hz).



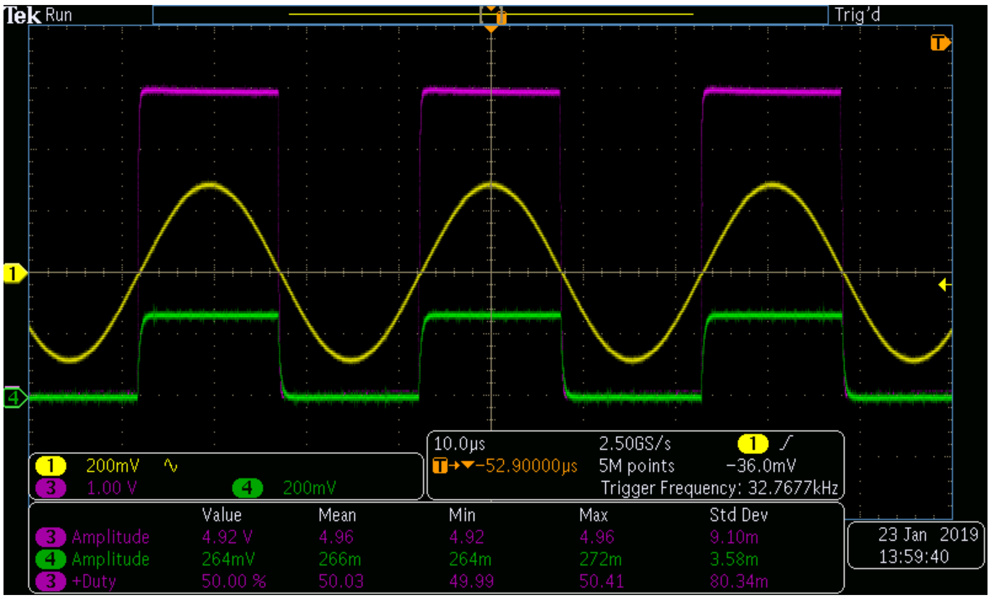
Trigger Frequency: 32.7777kHz

	Value	Mean	Min	Max	Std Dev
1 Peak-Peak	600mV	612m	600m	616m	5.22m
1 Min	24.0mV	17.8m	16.0m	24.0m	3.37m
1 Max	624mV	630m	624m	632m	3.73m
2 → 1 Phase	90.62 °	90.87	90.01	91.60	393.6m

Figure 4.9: Oscilloscope capture: Sine (yellow) and cosine (blue) wave at a frequency of 32.768 kHz + 10 Hz

4.4.2 Potentiometer Voltage Division

The 7-bit digital potentiometer is used in a voltage divider to convert the 5 V square wave from the DDS to a variable voltage level from 0 mV to 500 mV in steps of 3.9 mV for the current sink. Figure 4.10 shows the oscilloscope capture for the voltage division to 250 mV (50 mA current output), Figure 4.11 for the division to 400 mV (80 mA). The yellow trace corresponds to the DDS sine output (AC coupling) to demonstrate the pulse switching at zero-crossing. For the set voltage of 250 mV the value is off by 14 mV, for 400 mV division it is off by 8 mV. This is precise enough and can also be compensated in software.



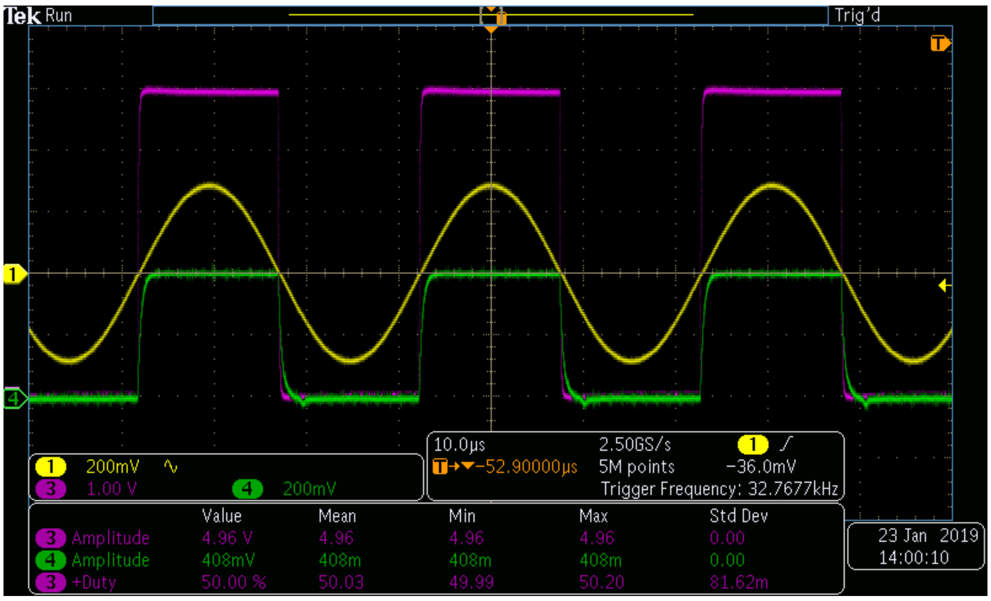
1 200mV ∿ 3 1.00 V 4 200mV

10.0µs 2.50GS/s 1 /
T→▼-52.90000µs 5M points −36.0mV
Trigger Frequency: 32.7677kHz

	Value	Mean	Min	Max	Std Dev
3 Amplitude	4.92 V	4.96	4.92	4.96	9.10m
4 Amplitude	264mV	266m	264m	272m	3.58m
3 +Duty	50.00 %	50.03	49.99	50.41	80.34m

23 Jan 2019
13:59:40

Figure 4.10: Oscilloscope capture: Voltage division to 250 mV (green) of the DDS square wave output (violet) with DDS sine (yellow) in AC coupling for reference



1 200mV ∿ 3 1.00 V 4 200mV

10.0µs 2.50GS/s 1 /
T→▼-52.90000µs 5M points −36.0mV
Trigger Frequency: 32.7677kHz

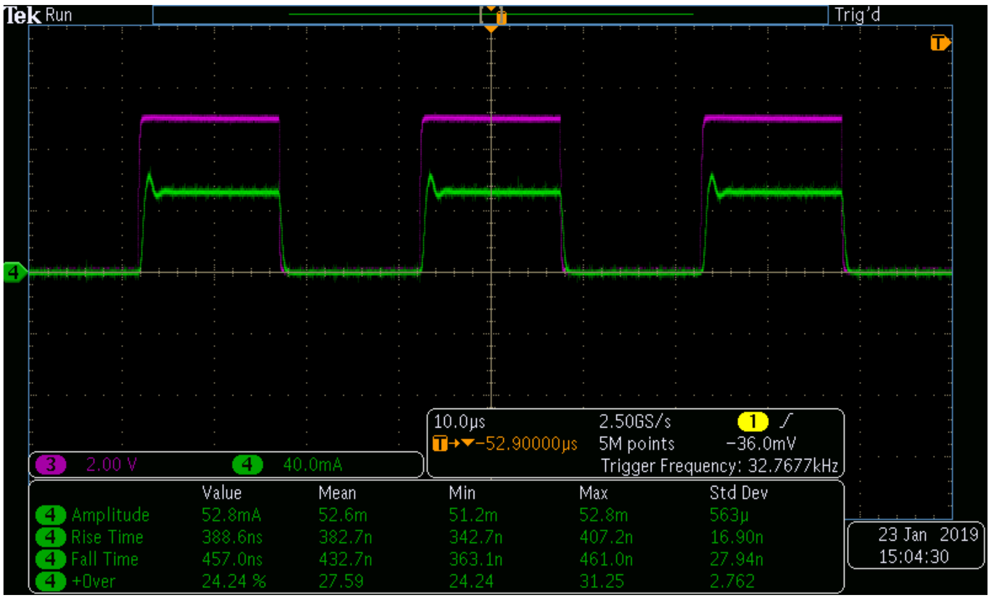
	Value	Mean	Min	Max	Std Dev
3 Amplitude	4.96 V	4.96	4.96	4.96	0.00
4 Amplitude	408mV	408m	408m	408m	0.00
3 +Duty	50.00 %	50.03	49.99	50.20	81.62m

23 Jan 2019
14:00:10

Figure 4.11: Oscilloscope capture: Voltage division to 400 mV (green) of the DDS square wave output (violet) with DDS sine (yellow) in AC coupling for reference

4.4.3 Current Sink Behaviour

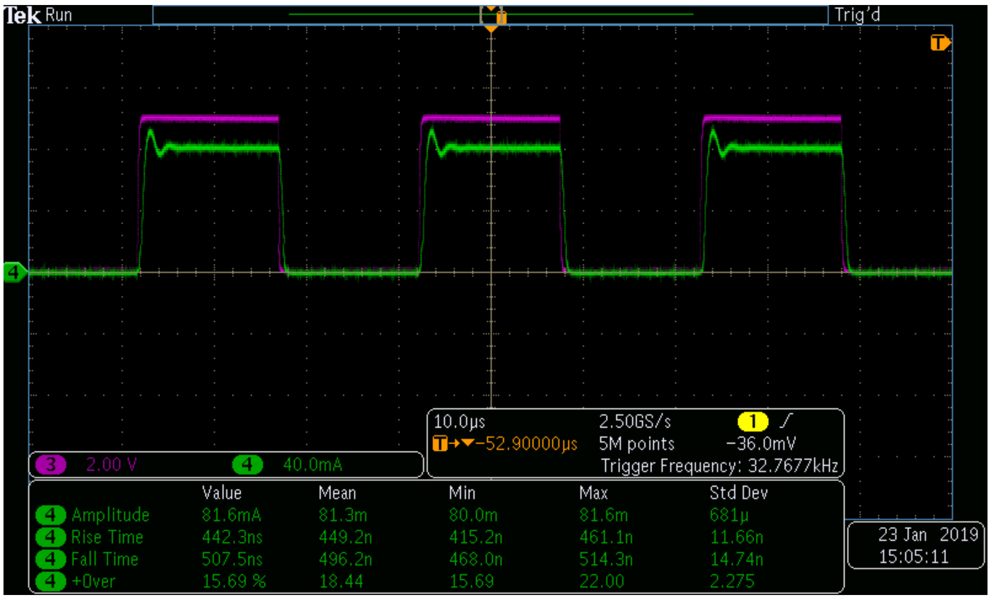
The square wave signal from the voltage divider is applied to the NMOS current sink, which regulates the current for the laser diode using a 5 Ω source resistance. Its electrical behaviour is illustrated in Figure 4.10 for a 50 mA output current (green) and in Figure 4.13 for a 80 mA output current. This is measured directly at the source resistor by setting the voltage to current relation on the oscilloscope to 5 V/A. The small overshoot of the output is a good compromise to the faster rise and fall times of less than 0.5 µs. The amplitude differs from the set value by 2 to 3 mA at a maximum controllable step size of 0.78 mA due to the digital potentiometer.



	Value	Mean	Min	Max	Std Dev
4 Amplitude	52.8mA	52.6m	51.2m	52.8m	563µ
4 Rise Time	388.6ns	382.7n	342.7n	407.2n	16.90n
4 Fall Time	457.0ns	432.7n	363.1n	461.0n	27.94n
4 +Over	24.24 %	27.59	24.24	31.25	2.762

Figure 4.12: Oscilloscope capture: The 52.8 mA current sink output in green with an overshoot of 24.24 % compared to the DDS square wave signal in violet

This current signal is within the acceptable tolerances for the QEPAS application since the exact current level is not too critical as long as the laser diode is not destroyed and the current level is constant over time. The fast rise and fall times on the other hand are advantageous in order to have a higher optical laser power.



	Value	Mean	Min	Max	Std Dev
4 Amplitude	81.6mA	81.3m	80.0m	81.6m	681µ
4 Rise Time	442.3ns	449.2n	415.2n	461.1n	11.66n
4 Fall Time	507.5ns	496.2n	468.0n	514.3n	14.74n
4 +Over	15.69 %	18.44	15.69	22.00	2.275

Figure 4.13: Oscilloscope capture: The 81.6 mA current sink output in green with an overshoot of 15.69 % compared to the DDS square wave signal in violet

4.4.4 Modifications

In the following, the most important modifications of the laser driver circuit are summarized and their effects are explained:

- The current sink feedback components are adapted for better current sink behaviour:
  R6, R28, R32, R33, R56, R57, R64, R65 → 1 kΩ
  R7, R29, R58, R66 → 150 Ω, C4, C17, C39, C43 → 1 nF
- Added capacitance at current sink output for better stability:
  C45, C46, C47, C48 → 1 nF
- AC coupling capacitors at DDS reference outputs removed: C33, C34 → 0 Ω
- Voltage supply for voice coil drivers connected to +5V_Driver to reduce noise influence of laser current
- Second SMA connector for LD_anode and LD_cathode at laser diode output because of an antenna effect of the coaxial cable

CHAPTER 5

Lock-In Amplifier

In the following, the design process of the lock-in amplifier is treated, which includes a brief explanation of the overall concept, the circuit design in KiCad, the LTspice simulation of the main lock-in circuit parts, the evaluation of the circuit as well as measurement results.

5.1 Introduction

The lock-in amplifier is based on the dual-phase lock-in amplifier concept and includes two identical lock-in amplifiers (LIA). One is fed with the weak signal input from the QTF and the sine reference signal (generated by the laser driver), the other one is lock-in amplifying the QTF signal with the cosine (90° phase-shifted) reference signal. These three signals are pre-amplified at first. The LIA itself is built with the balanced demodulator AD630 in synchronous demodulation configuration followed by low-pass filtering. The functionality of the LIA operation with the AD630 for the required frequency range around 33 kHz is supported by the datasheet [37] and additional papers [13,38], which also provide reference designs for a successful implementation. After the low-pass filtering the DC outputs of in-phase and quadrature signal (see Section 2.2.2) are digitalized by a 24-bit Analog-to-Digital Converter (ADC). Since this outputs can reach negative voltages as well, a bipolar ADC such as the ADS122C04 is required. The digital value of both signals can then be acquired with the microcontroller. For the power supply, a voltage of $\pm 6\,\mathrm{V}$ is considered, that gets converted to the required $\pm 5\,\mathrm{V}$ and the $\pm 2.5\,\mathrm{V}$ for the bipolar ADC reference by the use of four LDO regulators.

5.2 Circuit Design

The schematics of the important circuit parts will be discussed below. The pre-amplifier of the reference signal input and the lock-in amplifier are duplicated for the in-phase and quadrature signal path. The full circuit schematic and the PCB layout are attached in appendix B.

5.2.1 Pre-Amplifiers

Before being applied to the LIA, the reference sine and cosine signals provided by the laser driver as well as the measurement signal from the QTF are pre-amplified. In Figure 5.1 the pre-amplifier of the reference signals is shown. This is build up with the low noise operational amplifier OP27 [39] and has a fixed amplification of $A = 1 + \frac{R22}{R21} = 10$ set by the resistors **R21** ($1\,\mathrm{k\Omega}$), **R22** ($9\,\mathrm{k\Omega}$). Since the reference signals of the DDS are unipolar positive, the input of the OPA is AC coupled using the $1\,\mathrm{\mu F}$ capacitor **C280** and referred to ground by the $1\,\mathrm{M\Omega}$ resistor **R20**. The $10\,\mathrm{k\Omega}$ trim potentiometer realizes the offset voltage adjustment of the OP27. For the supply voltage decoupling the $10\,\Omega$ resistors **R37**, **R38** and the 100 nF capacitors **C29**, **C30** are inserted. This pre-amplifiers converts the 32-624 mV reference signal from the DDS to a $\pm 3\,\mathrm{V}$ signal.

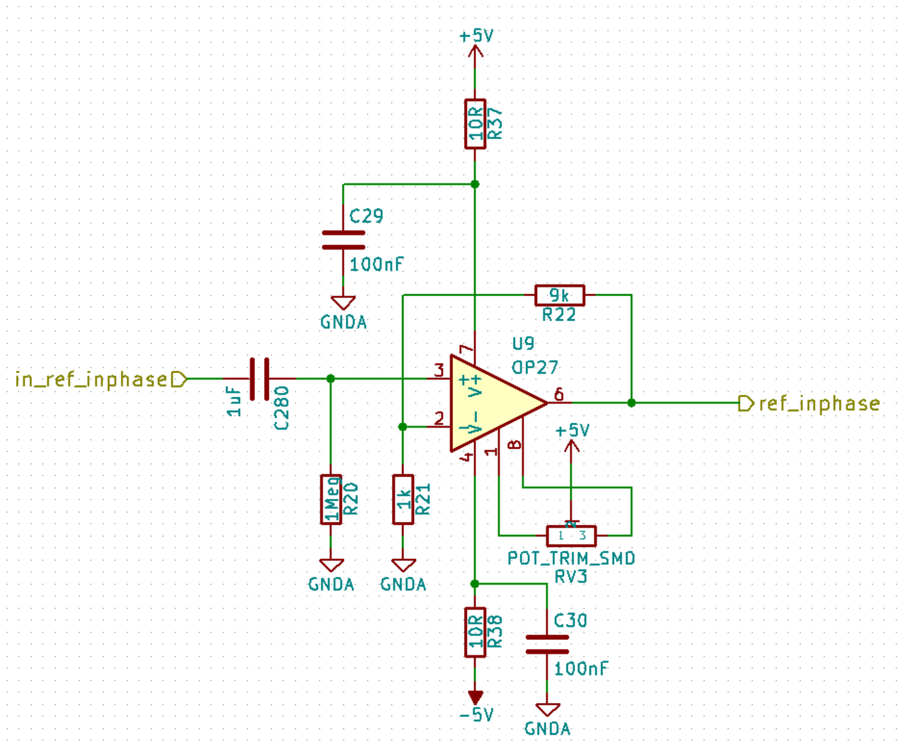


Figure 5.1: Schematic of the pre-amplifier for sine and cosine reference input

The pre-amplifier for the QTF signal is constructed with the OPA AD620 [40], which is a gain-controllable amplifier, to vary the gain according to the QTF signal amplitude. Therefore a fixed resistance **R16** (DNI) soldered between the **Rg** pins of the AD620 or in the other used case a manual 100 kΩ potentiometer **RV1** can set the gain. The pins **Rg1** and **Rg2** were conceived to be driven by a digital potentiometer similar to the one from the laser driver (Section 4.2.2), but failed in the realization due to the different voltage levels of the AD620 and the digital potentiometer. To calculate the resistance R_g for a certain gain G, the equation out of the datasheet is employed $R_g = \frac{49.4k\Omega}{G-1}$, or vice versa the gain is $G = 1 + \frac{49.4k\Omega}{R_g}$. The schematic can be seen in Figure 5.2. In order to bypass that pre-amplifier the jumpers **JP1** (connect 2 and 3) and **JP2** (open) are inserted. The input signal is AC coupled through the 1 µF capacitor **C15** and for the pre-amplification mode additionally referred to ground via **R14** (1 MΩ) and **R15** (10 Ω or 0 Ω). For the supply voltage decoupling the 10 Ω resistance (**R18**, **R19**) and 100 nF capacitance (**C25**, **C26**) are used for the 5 V and −5 V supply of the OPA.

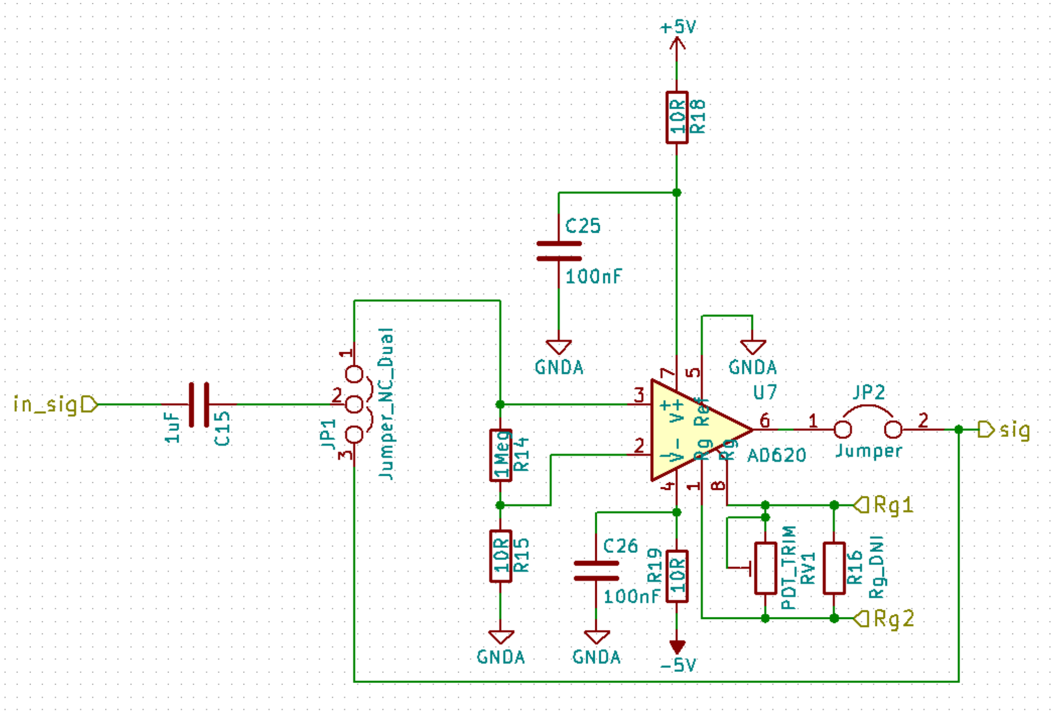


Figure 5.2: Schematic of the pre-amplifier for the QTF input signal, which can optionally be bypassed by the jumpers JP1, JP2

5.2.2 Lock-In Amplifier and Low-Pass Filters

The following in Figure 5.3 includes one of the two (dual-phase) lock-in amplifiers using the AD630 and the corresponding low-pass filtering (LPF) circuit.

Lock-In Amplifier

The balanced demodulator AD630 [37] in this configuration works in synchronous demodulation mode or lock-in amplification mode. The ±5 V supply voltage is decoupled by the resistors **R25**, **R26** and the capacitors **C16**, **C17**. The pins at the 10 MΩ resistors **R23**, **R24** and **R27** are supposed to be not connected, these resistors were inserted to may connect them to a stable potential (e.g. ground, this is also required for the simulation). For the offset adjustment the two precision 10 kΩ potentiometers are placed. The potentiometer **RV4** can adjust the common-mode (DC) offset, **RV5** is for the differential mode adjustment. The **Signal** input is the QTF signal, which is to be lock-in amplified, and the **Reference** input is the frequency reference signal from the DDS (sine or cosine wave). At pin **Vout** the phase-sensitive detector output is available according to the digital switching multiplier operation of the AD630 (compare Section 2.2.3), which is directly connected to the LPF.

Low-Pass Filters

The forth order low-pass filtering circuit is a combination of three passive RC-filters and one active low-pass filter (LPF) including a voltage division at **R33** (30 Ω), **R34** (20 Ω) to convert the maximum ±5 V output of the lock-in amplifier to a maximum input voltage of the ADC of 2 V at **Lockin1_out** ($\frac{R34}{R33+R34} \cdot 5V = 2V$). The cut-off frequencies $f_c = \frac{1}{2\pi RC}$ of the RC-LPFs and the active LPF in the order of their application and their corresponding components are listed in Table 5.1. For the inverting active LPF with a gain of $A = -\frac{R31}{R28+R29} = -1$ the amplifier OP27 is used with a 10 kΩ offset adjustment potentiometer **RV6** and a supply voltage decoupling by **R39**, **C31** and **R40**, **C32**.

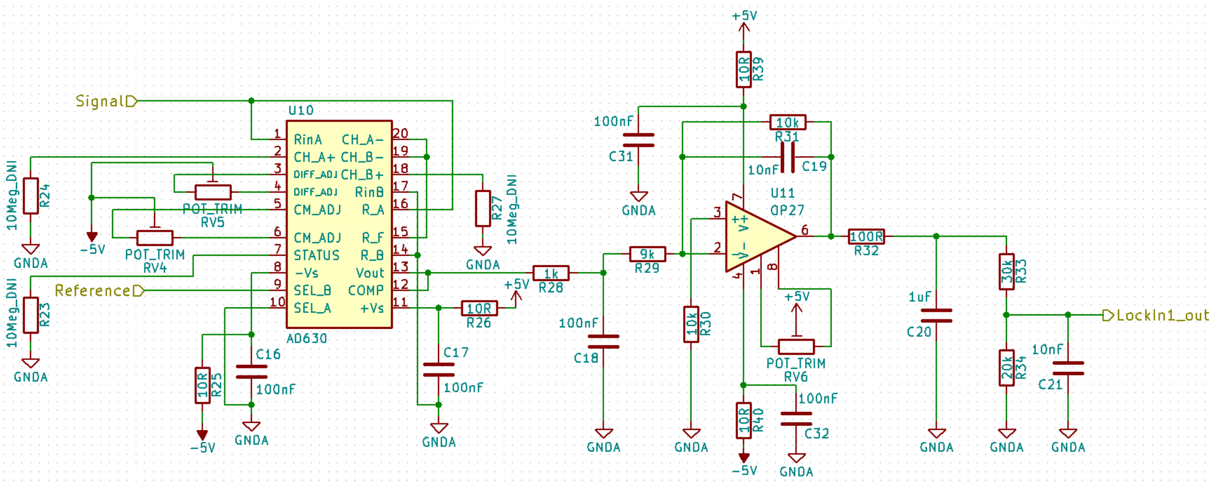


Figure 5.3: Schematic of the lock-in amplifier realized with the AD630 and a forth order passive/active low-pass filtering including a voltage divider for a 2 V maximum output voltage

order of application	LPF type	resistor	R value	capacitor	C value	cut-off frequency f_c
1	passive	R28	1 kΩ	C18	100 nF	1.5915 kHz
2	active	R31	10 kΩ	C19	10 nF	1.5915 kHz
3	passive	R32	100 Ω	C20	1 µF	1.5915 kHz
4	passive	R33	30 kΩ	C21	10 nF	0.5305 kHz

Table 5.1: Cut-off frequencies of the single LPFs with their corresponding components and values

5.2.3 Analog-to-Digital Converter

The DC output signal of the lock-in amplifier after low-pass filtering and voltage division is a ±2 V signal. That bipolar signal should be digitalized by an ADC. One of the few high-resolution ADCs, that is able to measure a single-ended bipolar signal, is the ADS122C04 [41] offering an I^{2}C interface and a 24-bit resolution. In Figure 5.4 its schematic is shown. The analog power supply is of a ±2.5 V voltage, whereas the digital part of the IC for the I^{2}C communication is supplied by the 5 V voltage **V_dig**. These voltages are decoupled by the resistor **R17** and the capacitors **C23**, **C24** and **C22**. For the opportunity to modify the I^{2}C address the resistors **R12** and **R13** are inserted. The external reference pins **REFP**, **REFN** are not used, since the IC has its own internal reference, and the data ready status pin $\overline{\textbf{DRDY}}$ is as well of no use. The interface clock **SCL** and data **SDA** pins are connected to the I^{2}C bus of the microcontroller.

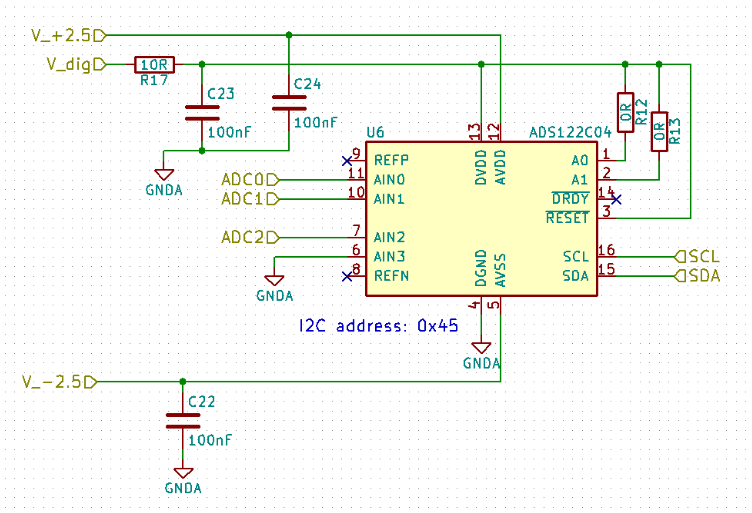


Figure 5.4: Schematic of the Analog-to-Digital Converter (ADC) for the bipolar in-phase and quadrature signal measurement

5.2.4 LDO Regulators and Ground Planes

The lock-in amplifier circuit is designed to be supplied by a bipolar ±6 V to ±7 V power supply. The four LDO regulators LT1761-5 [42], LT1964-5 [43], TPS72325 [44] and TPS78225 [45] convert this supply voltage to the required ±5 V and ±2.5 V voltage levels. Each LDO has a 1 µF input capacitance and a 10 µF output capacitance, depending on the typical application information of the regulators an additional bypass or feedback capacitor is added (**C6**, **C7**, **C9**). Figure 5.5 gives more detail about the schematic.

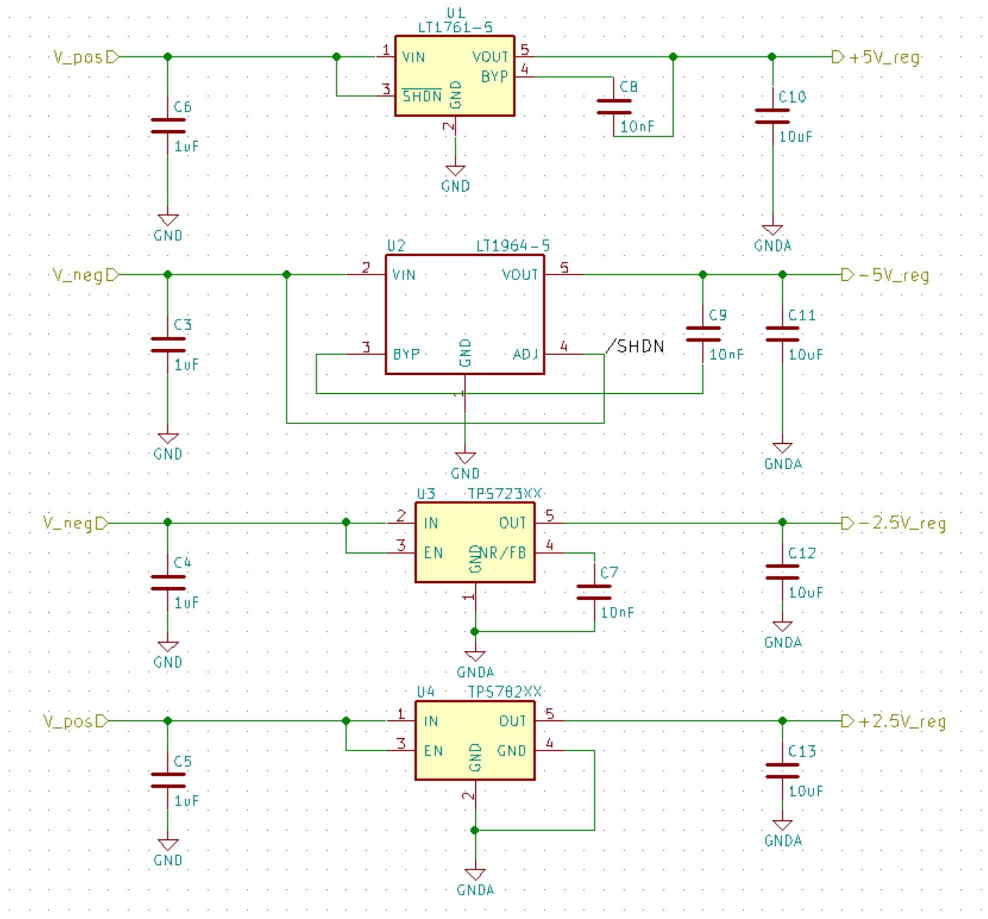


Figure 5.5: Schematic of the four low-dropout (LDO) regulators for the ±5 V and ±2.5 V voltage levels according to the application notes of their datasheets

Compared to the design of the laser driver, there is no power intensive circuitry or badly interfering digital device on this PCB. Therefore a top and bottom ground plane layout is efficient enough. The PCB layout is attached in appendix B.

5.3 Simulations

For the LTspice simulation of the lock-in amplifier circuit a macro-model of the AD630 from Analog Devices was used in combination with the low-pass filters according to the schematic of Figure 5.3. The reference input is set to a $\pm 3\,\mathrm{V}$ sinusoidal signal with a frequency of 32 kHz and the signal input is a $\pm 1\,\mathrm{V}$ sine signal with the same frequency. In Figure 5.6 the LIA behaviour is observed for a signal input with no phase shift respectively to the reference. The blue trace V(n014) after the phase-sensitive detector (PSD), which is the rectified input signal with gain -2, is low-pass filtered by the first passive LPF V(n011) followed by the active LPF V(int). The signal is further smoothened by the next LPF V(vtp) and the LPF with voltage division V(n016). The ripple-free DC signal V(n016) corresponds to the ADC input (5:2 voltage division in respect to V(vtp)).

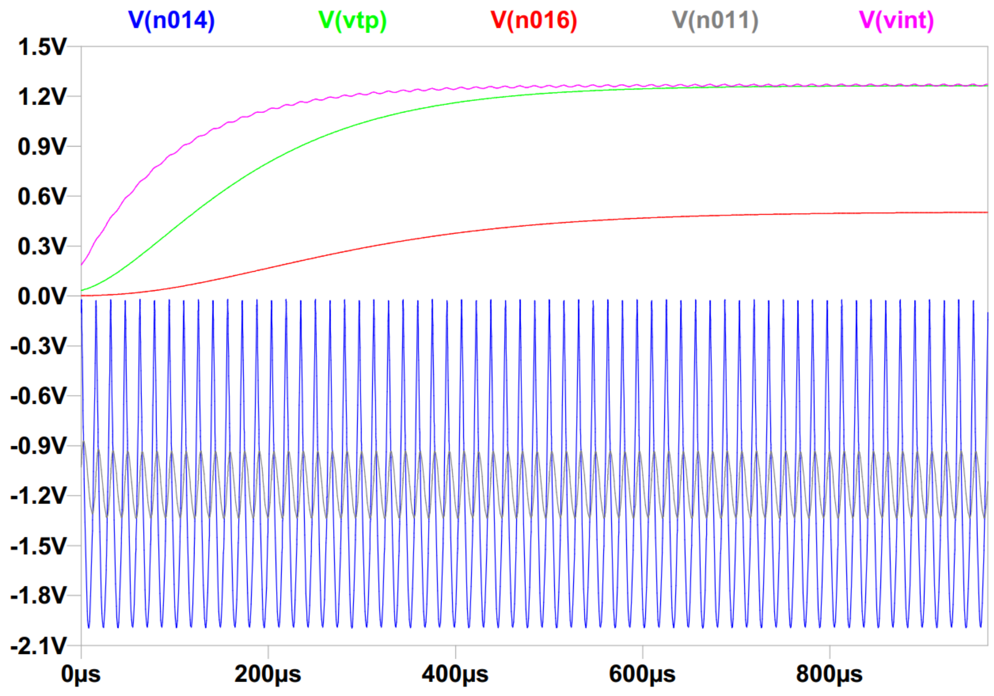
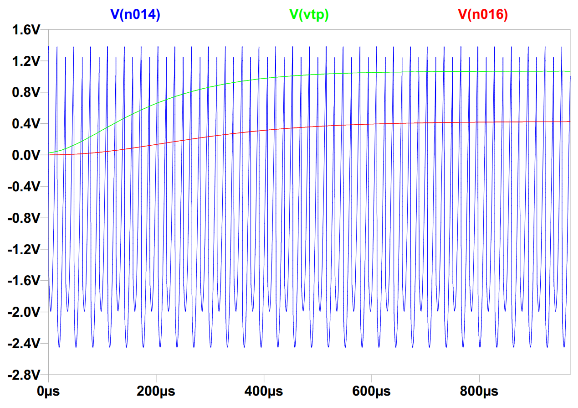


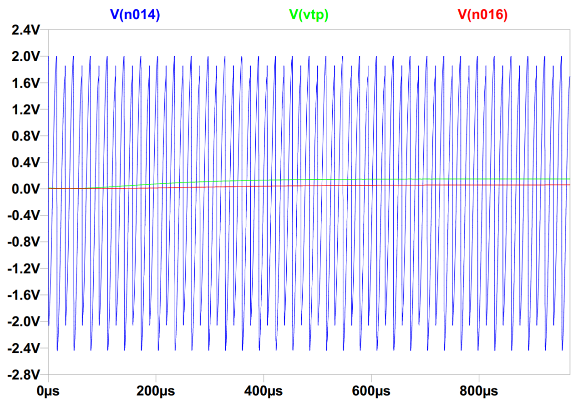
Figure 5.6: Simulation result: The rectified PSD output V(n014) is low-pass filtered by the 1st passive LPF V(n011), the active LPF V(int), the 3rd V(vtp) and the 4th LPF and voltage-divided V(n016). The input signal is of the same frequency and phase as the reference.

In the simulation above the expected behaviour of the AD630 is confirmed. The voltage level of the smoothed signal V(vtp) is about the calculated value of $\frac{2}{\pi} \cdot V_{pp} = 1.273\,\mathrm{V}$ and the one of the voltage-divided signal at $0.509\,\mathrm{V}$.

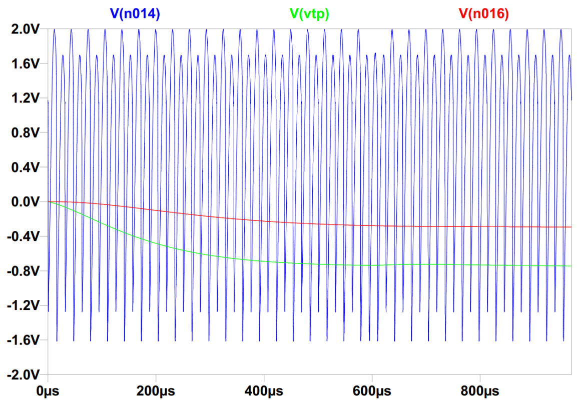
This simulation was repeated for different phases as shown in Figure 5.7. Compared to the evaluation measurements, the phase dependency with the factor of $cos(\Theta)$ is mathematically not as conclusive as there are deviations to calculated DC output values of V(n016). A possible cause might be the macro-model of the AD630, but the general working principle could be proven. For a phase of 45° the DC voltage output is decreased and at a phase shift of 90° the output is nearly zero. The behaviour for a 135° phase shift correlates with the negative response of 45° and the behaviour for 180° phase shift is the negative representation of the one of 0° phase.



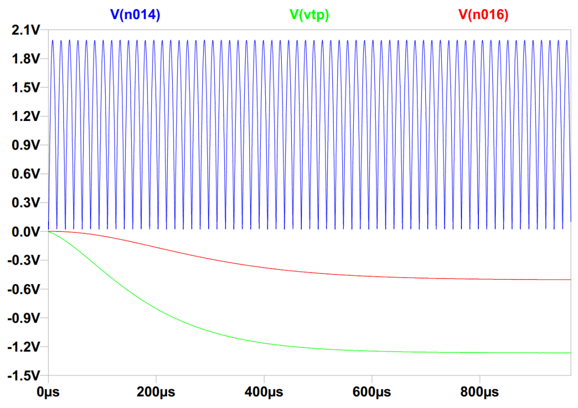
(a) LIA behaviour for a 45° phase-shifted input



(b) LIA behaviour for a 90° phase-shifted input



(c) LIA behaviour for a 135° phase-shifted input



(d) LIA behaviour for a 180° phase-shifted input

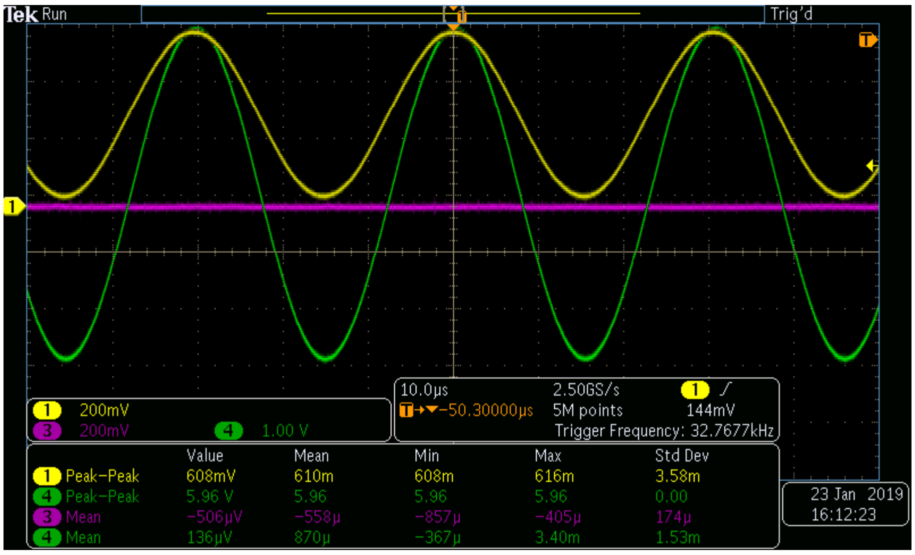
Figure 5.7: Simulation results: The PSD output V(n014) is low-pass filtered. The signal after the 3^{rd} LPF V(vtp) and after the 4^{th} LPF and the voltage divider V(n016) is shown. The input signal is of the same frequency, the phase is varied from 45° over 90° and 135° to 180°.

5.4 Evaluation and Verification Measurements

In this section the evaluation of the main parts of the lock-in amplifier circuit is discussed. Therefore the signal path is presented from the pre-amplifiers of the phase-sensitive detector through to the ADC signal and read-out. Furthermore, the overall performance of the LIA is presented concerning the frequency and phase response. The register configuration of the ADC is mentioned in Chapter 6. The important modifications of the LIA circuit during the evaluation phase are noted at the end of this chapter.

5.4.1 AC Coupling and Pre-Amplification

In order to convert the positive reference signals from the DDS to a bipolar signal with a higher amplitude an AC coupling and a 10-fold amplification is applied. This is realized by a 1 µF coupling capacitor and the amplifier OP27 with offset compensation using a trim potentiometer. The result is shown in the oscilloscope capture 5.8 for the sine reference signal. The DDS output (yellow) is AC coupled and amplified to a almost perfect ±3 V signal (green). The common-mode offset of this signal could be adjusted to less than 1 mV and the small deviation in amplitude is not an issue for the phase-sensitive detection with the AD630.



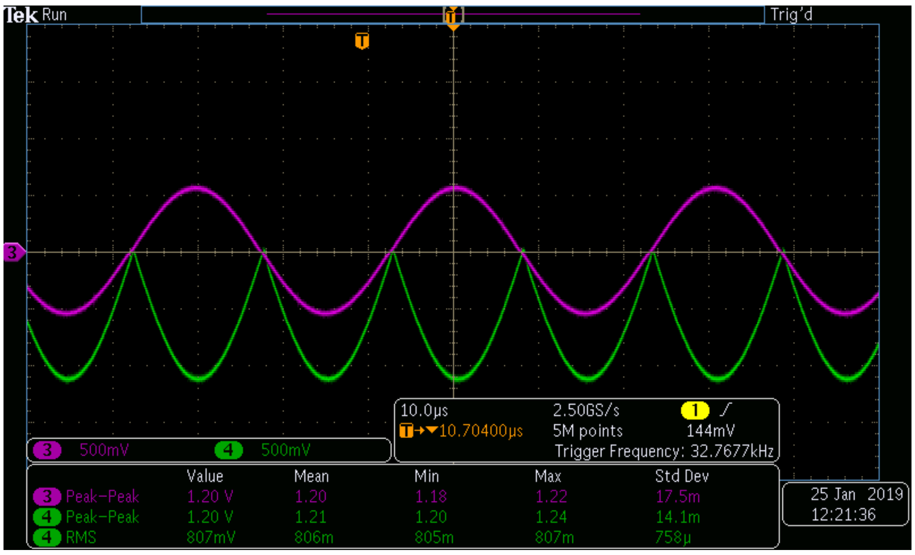
	Value	Mean	Min	Max	Std Dev
1 Peak–Peak	608mV	610m	608m	616m	3.58m
4 Peak–Peak	5.96 V	5.96	5.96	5.96	0.00
3 Mean	−506µV	−558µ	−857µ	−405µ	174µ
4 Mean	136µV	870µ	−367µ	3.40m	1.53m

Figure 5.8: Oscilloscope capture: The DDS reference signal (yellow) is AC coupled and pre-amplified to a ±3 V signal (green) with a small offset less than 1 mV

The performance of the pre-amplifier for the QTF signal is equally good with the exception that the gain is different and tuneable through the manual potentiometer **RV1**.

5.4.2 Phase-Sensitive Detector

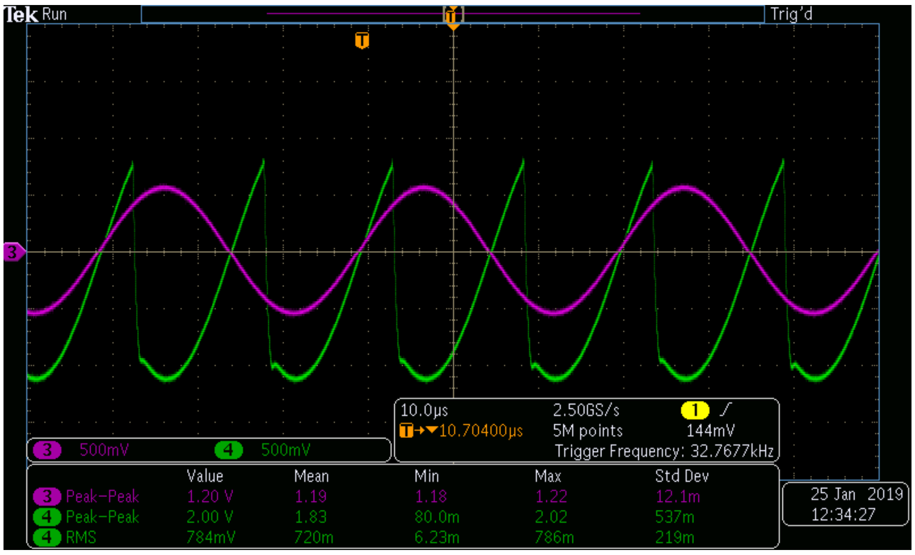
The phase-sensitive detector PSD is the centrepiece of the LIA. The AD630 is working as synchronous demodulator and is switching the polarity of the input signal according to the reference input (digital switching multiplier) with an additional gain of -2 in this configuration. In Figure 5.9 the oscilloscope capture is given for an input signal with the same frequency of 32.768 kHz and same phase as the reference. The PSD output signal (green) is the rectified version of the input signal (violet), switched at the zero-crossings and with -2 x gain. The amplitude is precisely doubled and the averaged value is proportional to the input signal's amplitude and phase ($cos(\Theta) = 1$), if it has the same frequency.



	Value	Mean	Min	Max	Std Dev
3 Peak-Peak	1.20 V	1.20	1.18	1.22	17.5m
4 Peak-Peak	1.20 V	1.21	1.20	1.24	14.1m
4 RMS	807mV	806m	805m	807m	758μ

Figure 5.9: Oscilloscope capture: The input signal of the PSD (violet) with the same frequency and phase as the reference is rectified at the zero-crossings and the PSD output (green) is proportional to its amplitude (gain -2 x).

If the phase is changing, e.g. to 45° like in Figure 5.10, the switching of the input signal occurs at a different point of the signal, the -2 x gain stays constant. The PSD output shape is not a rectification of the input signal any more and the mean value or integral over one period decreases, ideally by the factor of $cos(\Theta)$.

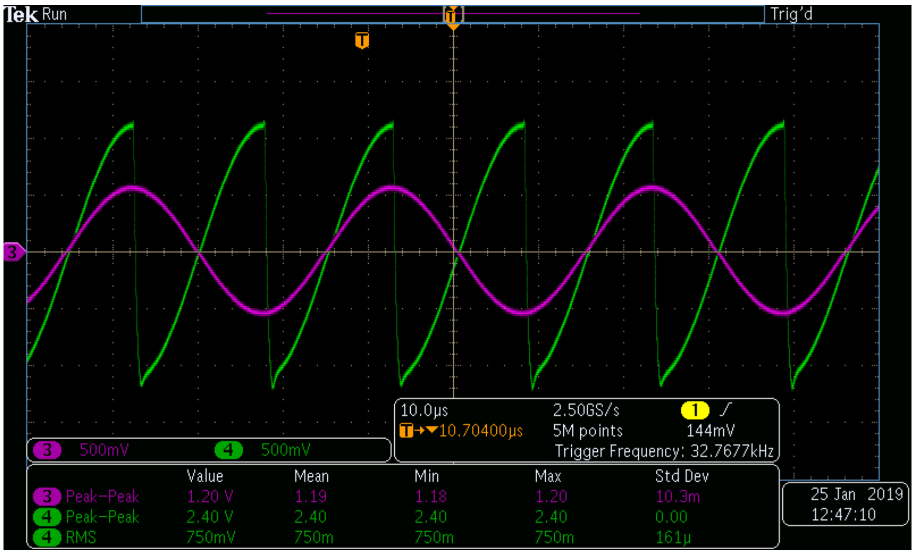


10.0µs   2.50GS/s   1 ∫
T→▼10.70400µs   5M points   144mV
Trigger Frequency: 32.7677kHz

	Value	Mean	Min	Max	Std Dev
3 Peak–Peak	1.20 V	1.19	1.18	1.22	12.1m
4 Peak–Peak	2.00 V	1.83	80.0m	2.02	537m
4 RMS	784mV	720m	6.23m	786m	219m

Figure 5.10: Oscilloscope capture: The input signal of the PSD (violet) with the same frequency as the reference, but phase-shifted by 45°, is switched at a different point and the mean value of PSD output (green) is decreasing.

By increasing the phase-shift to 90°, the input signal is now switched at its peak values. The PSD output signal is of twice the peak-peak value of the input and the mean value drops to zero. The characteristic signal curve of this case is shown in Figure 5.11. This is conform with the phase dependency of a PSD over the formula $cos(\Theta) = cos(90°) = 0$. For a dual-phase LIA the quadrature signal Y is in phase now ($sin(90°) = 1$) in that case and the in-phase signal X is zero (cf. Section 2.2.2 and Figure 2.7).



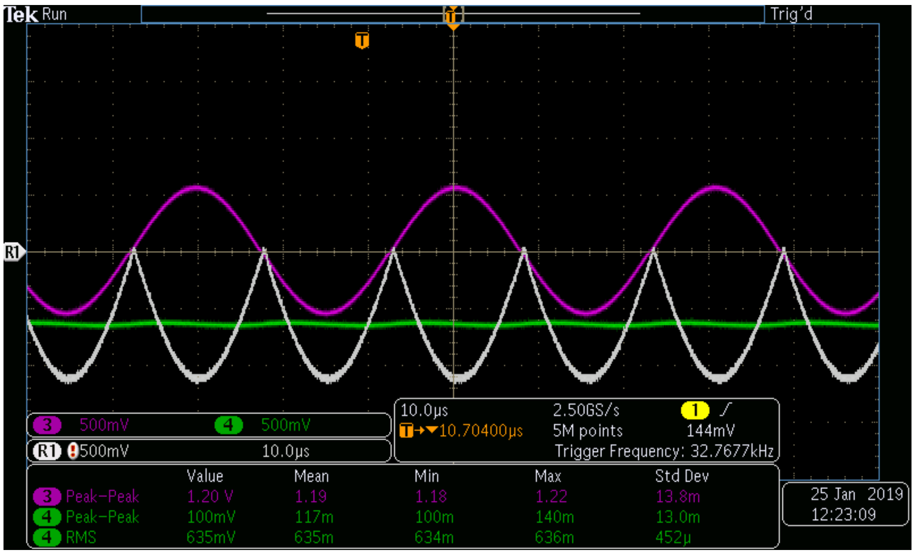
10.0µs   2.50GS/s   1 ∫
T→▼10.70400µs   5M points   144mV
Trigger Frequency: 32.7677kHz

	Value	Mean	Min	Max	Std Dev
3 Peak–Peak	1.20 V	1.19	1.18	1.20	10.3m
4 Peak–Peak	2.40 V	2.40	2.40	2.40	0.00
4 RMS	750mV	750m	750m	750m	161µ

Figure 5.11: Oscilloscope capture: The input signal of the PSD (violet) with the same frequency as the reference, but phase-shifted by 90°, is switched at its peak values and the average value of the PSD output (green) is zero.

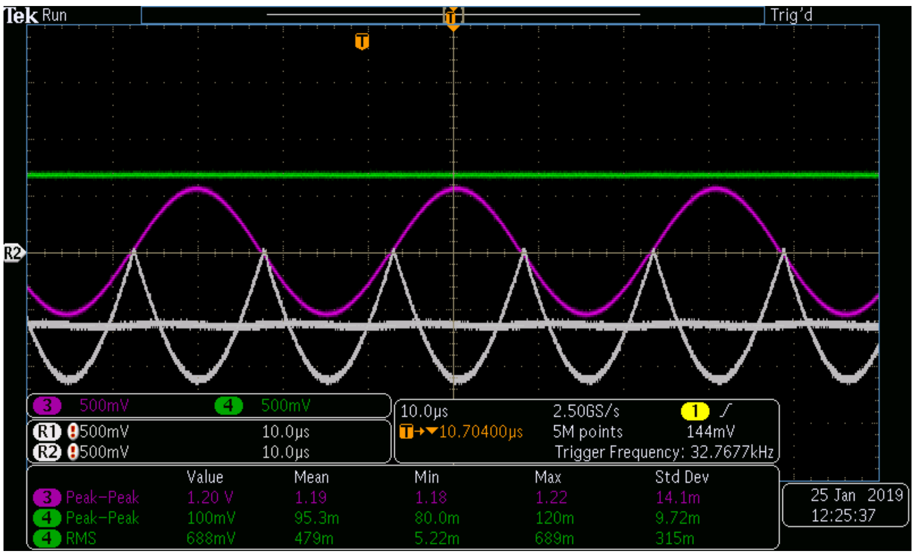
5.4.3 Low-Pass Filtering

The low-pass filtering with three passive and one active low-pass filters (LPFs) is applied to the PSD output, whose mean value is proportional to the input signal's amplitude and phase, in order to achieve an averaged DC signal for the ADC measurement. In the following examples the low-pass filtering is investigated with an input signal, which is in-phase and of the same frequency as the reference. After the first, passive LPF the signal appears like in Figure 5.12 (green). The PSD output in white is averaged, but the output still contains ripple. The second, active LPF in inverting amplification configuration further smooths the signal with a $-1\,\text{x}$ gain (Figure 5.13, green).



	Value	Mean	Min	Max	Std Dev
3 Peak–Peak	1.20 V	1.19	1.18	1.22	13.8m
4 Peak–Peak	100mV	117m	100m	140m	13.0m
4 RMS	635mV	635m	634m	636m	452µ

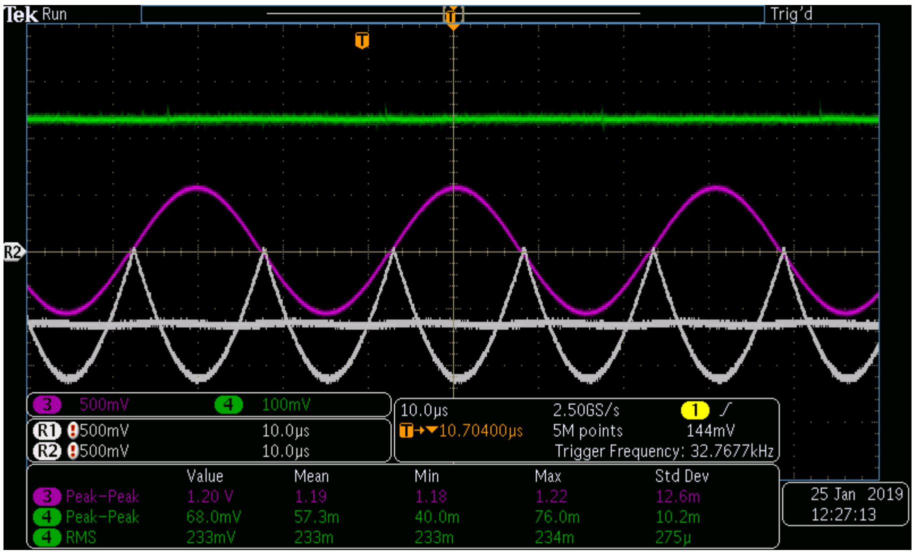
Figure 5.12: Oscilloscope capture: The 1st passive LPF averages the PSD output (white) to the green signal, the PSD input is the violet trace.



		Value	Mean	Min	Max	Std Dev
3	Peak-Peak	1.20 V	1.19	1.18	1.22	14.1m
4	Peak-Peak	100mV	95.3m	80.0m	120m	9.72m
4	RMS	688mV	479m	5.22m	689m	315m

Figure 5.13: Oscilloscope capture: The 2^{st} active LPF further smooths the PSD output with a gain of -1 to the green signal.

This output of the second filter is then applied to the third and forth, passive LPF and voltage divided by 5:2. The result is a ripple-free DC signal (Figure 5.14, green) with a maximum voltage of 2 V, which is connected to the ADC input.



		Value	Mean	Min	Max	Std Dev
3	Peak-Peak	1.20 V	1.19	1.18	1.22	12.6m
4	Peak-Peak	68.0mV	57.3m	40.0m	76.0m	10.2m
4	RMS	233mV	233m	233m	234m	275µ

Figure 5.14: Oscilloscope capture: After the 3^{rd} and 4^{th} LPF (passive) and a 5:2 voltage division the ADC input signal (green) is available.

The overall conversion factor from LIA input amplitude to ADC input is calculated to 0.509 and confirmed by the simulation results. Whereas the actual measured conversion factor is 0.389 without taking the pre-amplifier gain into account and can be assumed constant (e.g. for calculations). This deviation is caused by the losses during low-pass filtering. One compensatory method to reach a specific conversion factor would be to trim the gain of the signal pre-amplifier.

5.4.4 ADC Evaluation

The analog-to-digital converter ADS122C04 is a 24-bit ADC capable of measuring bipolar signals, meaning positive voltages as well as negative ones. Although it offers a internal voltage reference, the external $\pm 2.5\,\mathrm{V}$ supply is used, since it could not get to work. In application a voltage resolution of 298 nV at a maximum sampling rate of 2000 SPS can be achieved. The restriction from the microcontroller limit this to a float data type (6 decimal digits) in resolution and a sampling of 100 SPS with a measurement delay between the two channels of 1 ms.

In the evaluation phase the accuracy of the ADC readout was as precise as the multimeter or oscilloscope. Taking the measurement from before out of Figure 5.14 for example, the readout value was 233.321 mV compared to a 233 mV measured on the oscilloscope and hence way more accurate.

5.4.5 Frequency Response of the Lock-In Amplifier

For the evaluation of the frequency response of the LIA a measurement routine is implemented using the two DDS from the laser driver to generate the frequency signals. Since there are only two of the DDS available, the dual-phase LIA is operated in single-phase operation. The reference signal for the in-phase and quadrature path is therefore the same signal with with the QTF's resonance frequency of 32.768 kHz generated by the first DDS. As input signal the second DDS generates a frequency, that is varied in a certain frequency range. To obtain the narrow bandwidth response, a frequency sweep from 32.75 kHz to 32.79 kHz with a step resolution of 0.1 Hz is chosen. The microcontroller performs this sweep, takes an ADC readout of in-phase and quadrature channel every 10 ms and sends the results to the MATLAB script on the PC. In this measurement the sampling rate is set to 100 SPS, according to the 10 ms time routine, and the repetition rate for each frequency step to a 100 Samples, that are averaged later on. This equals to an integration time of 1 s per measurement point. The resulting magnitudes of the in-phase ($|V_{inphase}|$) and quadrature channel ($|V_{quadrature}|$) for this LIA narrow band response are plotted in Figure 5.15.

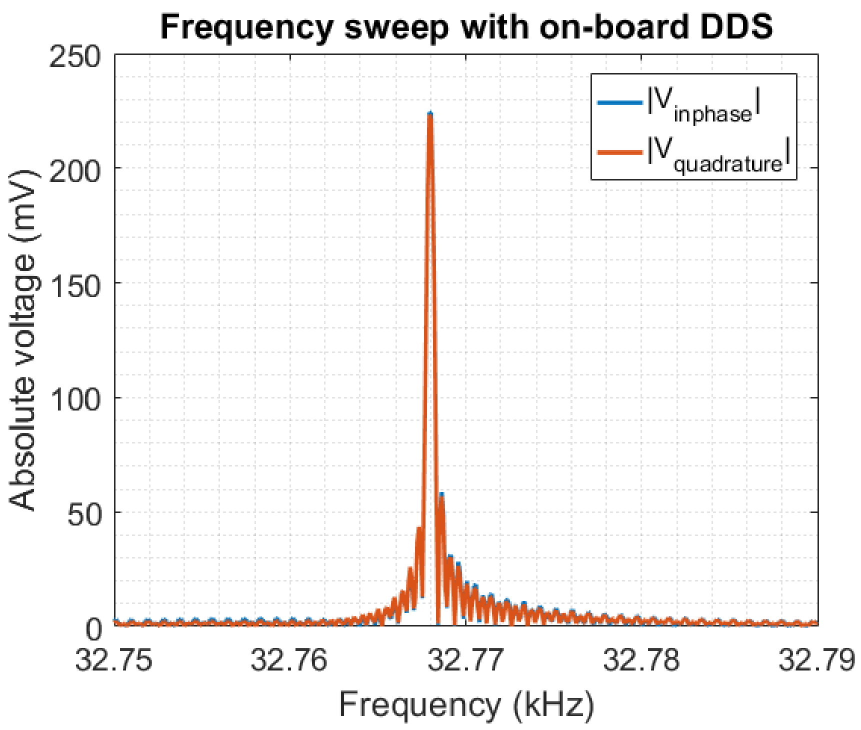


Figure 5.15: Narrow band frequency response of the LIA with a locked frequency of 32.768 kHz in the range from 32.75 kHz to 32.79 kHz

The result shows a precise frequency selectivity at the locked frequency and a near-identical behaviour of the in-phase and quadrature channel. Its narrow FWHM (Full Width at Half Maximum) bandwidth of 0.6 Hz and fast roll-off behaviour are excellent properties for a LIA. In the frequency band of 3 Hz around the locked frequency the signal attenuation reached already to less than 10 % of the peak value. To express the sharpness of the LIA in the term of Q-factor, this can be calculated to $Q = \frac{f_r}{\Delta f} = 54613.33$ with f_r as resonance frequency and Δf as FWHM bandwidth.

Since the lock-in amplifier is build up with phase-sensitive detectors based on the principle of a digital switching multiplier, which let also pass through the odd harmonics (square wave multiplication), this behaviour is also examined. For this reason a broader frequency sweep was performed for a rough performance analysis and multiple frequency sweeps around the odd harmonics from the 1^{st} up to 17^{th} with a higher resolution. The odd harmonics sweeps are combined in one plot and the result for the in-phase ($|V_{inphase}|$) as well as quadrature channel magnitude ($|V_{quadrature}|$) with the same settings as before and the 32.768 kHz as reference frequency is illustrated in Figure 5.16. Points between the several harmonics are ignored and can be expected to be very low compared to the peaks based on the insights of the first measurement with a broader spectrum.

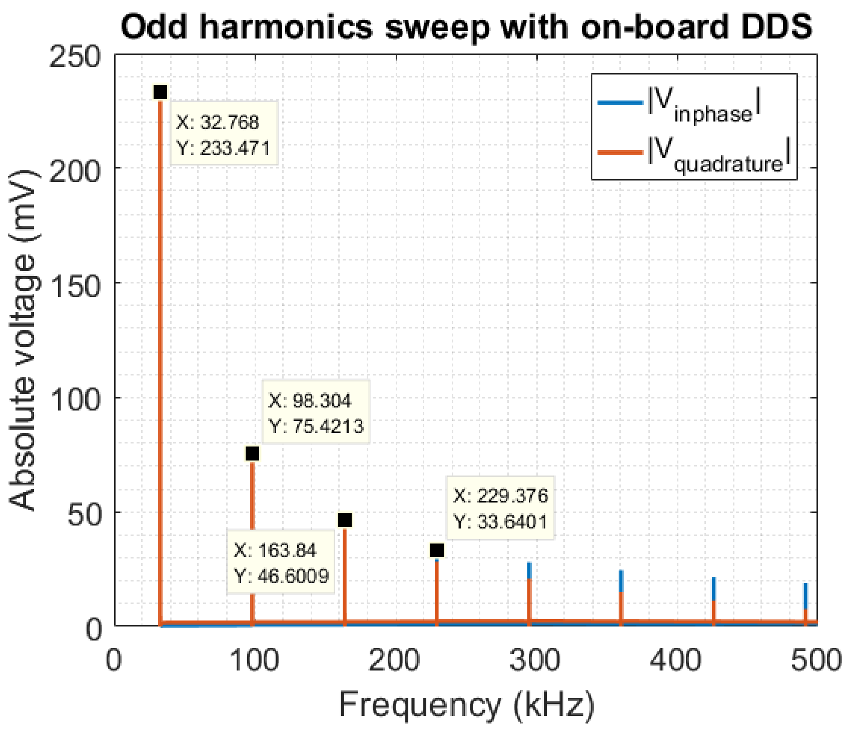


Figure 5.16: Broadband frequency analysis of the LIA with a locked frequency of 32.768 kHz at the odd harmonics from the 1^{st} to the 17^{th} harmonic

It can be seen, that the odd harmonics indeed contribute to the frequency response. At the resonance frequency the amplitudes of $|V_{inphase}|$ and $|V_{quadrature}|$ are 233.47 mV, for the 3^{rd} harmonic it is 75.42 mV (compared to a calculated one third $\frac{1}{3}$ of 77.82 mV), the 5^{th} is at 46.6 mV (calc. $\frac{1}{5}$: 46.69 mV) and the 7^{th} at 33.64 mV (calc. $\frac{1}{7}$: 33.35 mV). Amplitudes of higher odd harmonics are slightly larger than the calculated values, whereas the in-phase component $|V_{inphase}|$ is more off than the quadrature signal $|V_{quadrature}|$. This condition has to be considered in the application for weak signals with a broadband noise. An improvement to avoid this contribution of odd harmonics is a bandpass (or low-pass) filtering at the input of the LIA in the range of the observed frequency.

5.4.6 Phase Response of the Lock-In Amplifier

Similar to the frequency response before, the phase response is investigated using the second DDS to generate a signal with the same resonance frequency of 32.768 kHz as the reference, but with another phase. This is varied from 0° to 360° at a step size of 0.25° in an implemented self-test routine. The ADC settings therefore stay the same at a 100 SPS, an averaging of a 100 Samples and a resulting integration time of 1 s.

Both LIA outputs, $V_{inphase}$ and $V_{quadrature}$, are presented in Figure 5.17 with regards to their phase relation to the reference signal.

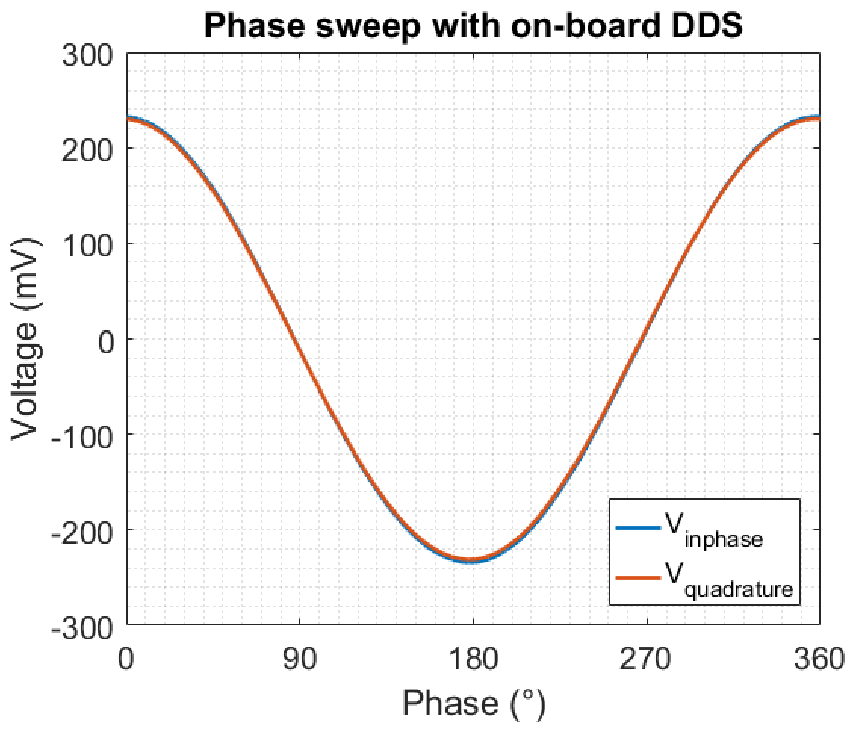


Figure 5.17: Phase response of the LIA at the frequency of 32.768 kHz in the range from 0° to 360° phase in respect to the reference signal with a step size of 0.25°

The measured output signals of the in-phase and quadrature component do again match well. The response to the phase change is a near-ideal cosine dependency, which is mathematically expressed through $V = V(\Theta = 0) \cdot cos(\Theta)$. The maximum deviation to the mathematical ideal cosine is $\pm 12.5\,\text{mV}$ or $\sim 5\,\%$ at a phase of 90° and 270°, which corresponds to a maximum phase error of around ±3°. This constant phase error can be corrected mathematically in software (e.g. lookup table, polynomial correction, etc.) for a higher phase precision.

The evaluation of the maximum phase error at 90° phase is demonstrated in Figure 5.18. The LIA signals are tagged with an amplitude of $-12.51\,\text{mV}$ compared to an ideal value of $0\,\text{mV}$. The ideal $cos(\Theta)$ at this offset has a phase between 93.00° and 93.25° and consequently the phase error in this case is $-3.1°$. For a phase of 270° the maximum error is a positive one with approx. $+3°$.

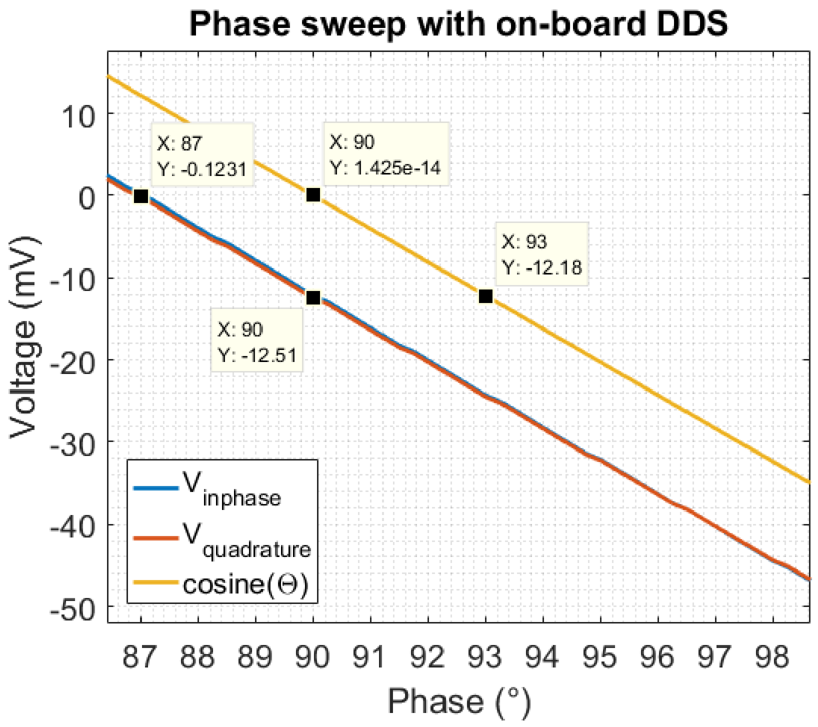


Figure 5.18: Phase error evaluation around 90° wit the two LIA outputs $V_{inphase}$, $V_{quadrature}$ and the ideal cosine phase dependency $cosine(\Theta)$

5.4.7 Modifications

The most important modifications applied to the lock-in amplifier circuit during the evaluation phase are listed and explained below:

- An AC coupling at the signal and reference inputs is inserted/replaced by a CR element consisting of a 1 µF capacitance in series and a 1 MΩ resistor to ground, in order to have a AC coupling referred to ground (floating potential before):
  C15 → C15 = 1 µF in series, R140 = 1 MΩ to GND,
  R20 → C280 = 1 µF in series, R20 = 1 MΩ to GND,
  R35 → C281 = 1 µF in series, R35 = 1 MΩ to GND
- The low-pass filters after the AD630 were adopted to lower cut-off frequencies according to Table 5.2.
- The analog input channel 3 (**AIN3**) of the ADS122C04 was connected to ground to have a 0 V reference potential for the single-ended bipolar signals of the LIA outputs.
- The ADC reference voltage was reconfigured from the internal reference to the ±2.5 V analog supply voltage.

order of application	LPF type	resistor	R value	capacitor	C value	cut-off frequency f_c
1	passive	R28	1 kΩ	C18	100 nF	1.5915 kHz
2	aktive	R31	10 kΩ	C19	1 µF	15.915 Hz
3	passive	R32	10 kΩ	C20	1 µF	15.915 Hz
4	passive	R33	30 kΩ	C21	1 µF	5.305 Hz

Table 5.2: Modifications of the cut-off frequencies of the single LPFs with their corresponding components and values

CHAPTER 6

Interface

This chapter covers the interface between the developed circuits and the microcontroller and moreover the communication from the microcontroller to the PC using MATLAB. For this project the Arduino Uno is used as microcontroller and the firmware is programmed in the Arduino IDE software with its available libraries. The MATLAB communication is realized via COM port over USB.

6.1 Introduction

In the final solution a specific command for each of the measurement types is send from MATLAB to the microcontroller, that handles the current measurement routine including the configuration/communication to the IC devices, and returns the ADC read-outs to the MATLAB script. The following sections cover the configuration of the ICs (DDS, digital potentiometers, ADC) and the corresponding implemented functions. Furthermost, the microcontroller boot-up setup, the available measurement modes, which are performed in a timer routine on the Arduino, as well as the serial communication including the command parsing and the MATLAB communication script, which also saves the acquired data in a .txt file, are explained. The Arduino firmware code is split up into a hardware description code in standard C as an include library (hw_driver_serialOnly.h, Appendix C) and a serial communication and measurement routine code (sweep_fre_phase_serialOnly.ino, Appendix D).

6.2 IC Configuration

This section should give an overview of the correct configuration of the IC registers. Moreover, the dedicated, implemented functions are given and discussed. All of this is realized in the hardware description code (hw_driver_serialOnly.h, Appendix C).

6.2.1 DDS Configuration and related Functions

The AD9834 contains a 16-bit control register, which defines the operation of the DDS, as well as two frequency FREQ0, FREQ1 and two phase registers PHASE0, PHASE1. The first two bits DB15, DB14 (or the first three in the case of phase control) of each 16-bit register determine the register address according to Table 6.1 and the bits DB13 to DB0 the particular register value (DB11 to DB0 for the 12-bit phase value). In Table 6.2 the chosen configuration for the control register bits is presented and each bit's function explained.

DB15	DB14	DB13...DB0		
0	0	CONTROL bits		
0	1	14 FREQ0 register bits		
1	0	14FREQ1 register bits		
DB15	DB14	DB13	DB12	DB11...DB0
1	1	0	X	12 PHASE0 bits
1	1	1	X	12 PHASE1 bits

Table 6.1: Addressing of the control, frequency and phase registers [30]

To write to the 28-bit frequency registers as a complete word beginning with the 14 LSBs followed by the 14 MSBs, the consecutive write operation is enabled (B28 = 1). The address bits DB15 and DB14 thereby stay the same. Since the reset function is used with the hardware pin to achieve a synchronisation of both DDS, the frequency and phase register are defined by the digital pins FSELECT, PSELECT and hence select the FREQ0 and PHASE0 register.

Control Register = 0x2228			
Bit	Name	Value	Description
DB13	B28	1	Allows consecutive write operation to load a complete word into the frequency registers, if B28 = 1.
DB12	HLB	0	This bit select the MSBs (HLB = 1) or LSBs (HLB = 0) of the frequency register to be written to, if no consecutive write operation is required. Therefore DB28 has to be set to 0.
DB11	FSEL	0	FSEL in combination with PIN/SW defines which frequency register is used for the frequency generation, FSEL = 0 for FREQ0 and FSEL = 1 for FREQ1.
DB10	PSEL	0	PSEL in combination with PIN/SW defines which phase register is used for the frequency generation, PSEL = 0 for PHASE0 and PSEL = 1 for PHASE1.
DB9	PIN/SW	1	Selects whether the functions of frequency and phase register selection, the reset of internal registers and power down is controlled by the hardware pins or the control bits. PIN/SW = 1 is set for control by the hardware pins and PIN/SW = 0 for the control bits.
DB8	RESET	0	RESET = 1 resets the internal registers to 0, which corresponds to an analog output of midscale, RESET = 0 for no reset.
DB7	SLEEP1	0	SLEEP1 = 1 disables the internal MCLK (DAC output remains at present value), SLEEP1 = 0 enable MCLK and accumulation.
DB6	SLEEP12	0	Powers down the on-chip DAC for SLEEP12 = 1, otherwise it is active.
DB5	OPBITEN	1	For OPBITEN = 1 the square wave output at SIGN BIT OUT is enabled.
DB4	SIGN/PIB	0	This bit controls the output function of SIGN BIT OUT pin. SIGN/PIB = 1 is set to use the comparator input, for SIGN/PIB = 0 the MSB of the DAC data is connected to SIGN BIT OUT.
DB3	DIV2	1	DIV2 controls whether the SIGN BIT OUT output is the MSB (DIV2 = 1) or MSB/2 (DIV2 = 0) of the DAC data.
DB2	Reserved	0	Must always be set to 0
DB1	Mode	0	The MODE bit controls the output function of the IOUT/IOUTB pin. If it is set to 1 the output is a triangle and if it is set to 0 the SIN ROM converts the output to a sinusoidal.
DB0	Reserved	0	Must always be set to 0

Table 6.2: Description of the control register and the set bit values [30]

In order to write a specific frequency value into the FREQ0 register the addressing of Table 6.1 is used and the 28-bit value split into 14 LSBs FREQREG[13:0] and 14 MSBs FREQREG[27:14] and send to the DDS in that order. The relation between bit value and frequency is stated as $frequency = \frac{f_{MCLK}}{2^{28}} \cdot FREQREG$, where f_{MCLK} is the master clock with 8 MHz. The phase is loaded into the 12-bit PHASE0 register with the phase relation $phase = \frac{2\pi}{2^{12}} \cdot PHASEREG$ in a single write operation.

The implemented functions for the DDSs configuration and operation are listed below with a short description. For the SPI communication on the Arduino a write-only implementation is realized with the function void spiWrite(int CS_pin, int len, byte ∗sequence), where CS_pin is the chip select of the SPI device, len the length of the write sequence and sequence the byte array of the command. The used Arduino function void digitalWrite(int pin, bool value) sets a digital pin of the Arduino to HIGH or LOW. This function is used for the synchronization of the DDS by the reset pin.

void initDDS():
Initializes both DDSs by writing the mentioned configuration to the control register and putting them into reset mode.

void enableDDS():
Enables both DDSs by disabling the reset mode and activating the output signals.

void disableDDS():
Puts the DDSs into reset mode, which results into midscale value at the outputs.

void DDSI_SetFreqPhase(float freq, float phase):
This function calculates the register values for the frequency and phase register from the inputs freq in Hz and phase in °, followed by a write operation to these registers and an additional write of the control word. The function is assigned to the in-phase DDS. It is recommended to put the DDS into reset mode during that write operation using the void disableDDS() and void enableDDS().

void DDSQ_SetFreqPhase(float freq, float phase):
This is the same function as void DDSI_SetFreqPhase(float freq, float phase), but dedicated to the quadrature DDS.

void DDS_SetFreqInQuadrature(float freq):
For setting up both DDSs with the same frequency in quadrature operation, meaning with a 90° phase shift, this function is used. It includes the disabling and enabling of the DDSs during the communication.

6.2.2 Digital Potentiometer Functions

The dual digital potentiometer (rheostat) MCP4652 with a maximum resistance of 10 kΩ (MAX_RESIST = 10000) is used to control the laser current. This IC does not need a configuration, however, the resistance value should be set to zero at boot-up since it powers up with a midscale value. The following functions can be used to set a specific resistance value or a certain current for the laser driver output:

> bool setMCP4652_Resistance(byte address, int channel, unsigned long resist):
> This function sets the resistance value (resist) in Ω of channel 0 or 1 of the MCP4652 with the specified I^2C address and returns whether the I^2C communication was successful or a boolean false if resist exceeds the MAX_RESIST value.
>
> bool setLaser1_I(float mA):
> In order to set a current on the laser stage 1, the function calculates the resistance of the digital potentiometer from the input current (mA) in mA and addresses the corresponding potentiometer and channel.
>
> bool setLaser2_I(float mA):
> Same function as bool setLaser1_I(float mA) for the laser stage 2.

6.2.3 ADC Configuration, Conversion and Read-Out Functions

As listed in Table 6.3, the ADS122C04 offers six control commands. For the register read and write commands the register address **rr** is required. These can access the four 8-bit configuration registers, that include a variety of settings. The register map for these is shown in Table 6.4, whereof just the first two are of importance in our case.

Command	Description	Command Byte(1)
RESET	Resets the device	0000 011x
START/SYNC	Start or restarts conversion	0000 100x
POWERDOWN	Enters standby mode	0000 001x
RDATA	Read data command	0001 xxxx
RREG	Read register at address rr	0010 rrxx
WREG	Write register at address rr	0100 rrxx

Table 6.3: Command definition, (1) Symbols: rr = register address, x = don't care [41]

Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
00h	MUX[3:0]				GAIN[2:0]			PGA_BYPASS
01h	DR[2:0]			MODE	CM	VREF[1:0]		TS
02h	DRDY	DCNT	CRC[1:0]		BCS	IDAC[2:0]		
03h	I1MUX[2:0]			I2MUX[2:0]			0	0

Table 6.4: Configuration register map [41]

The standard setting is a continuous conversion of channel 0 (AIN0) referred to channel 3 (AIN3), which is connected to ground (GND), with a gain of 1 and a bypassed programmable gain amplifier (PGA) at a data rate of 2000 SPS in turbo mode. Except the internal reference voltage, that is set to supply voltage, the other configuration are as default. The applied configuration of the ADS122C04 is shown in detail in Table 6.5. In all use cases the last two configuration registers (address 02h, 03h) are kept on default values.

Bit	Name	Value	Description
Configuration Register 0 (address = 00h, default = 00h)			
7:4	MUX[3:0]	0010b	Configures the input multiplexer to positive input AIN$_P$ = AIN0 and negative input AIN$_N$ = AIN3 = GND.
3:1	GAIN[2:0]	000b	Sets the gain setting to Gain = 1 (default).
0	PGA_BYPASS	1b	Bypasses/disables the programmable gain amplifier. Only gain setting 1 x, 2 x and 4 x are allowed, if PGA is disabled.
Configuration Register 1 (address = 01h, default = 00h)			
7:5	DR[2:0]	110b	Controls the data rate and sets this to 2000 samples per sencond (SPS) in turbo mode (MODE = 1h). Conversion time: typ. 0.51 ms
4	MODE	1b	Enables the turbo mode.
3	CM	1b	Continuous conversion mode is activated.
2:1	VREF[1:0]	10b	Voltage reference selection is set to analog supply (AVDD - AVSS).
0	TS	0b	Disables the internal temperature sensor mode.
Configuration Register 2 (address = 02h, default = 00h)			
7	DRDY	0b	Read-only flag for a new conversion result toggles to 1 if data is ready.
6	DCNT	0b	This bit enables the conversion counter (disabled = 0b).
5:4	CRC[1:0]	00b	Allows a data integrity check if enabled (disabled = 00b).
3	BCS	0b	10 µA burn-out current source is set off.
2:0	IDAC[2:0]	000b	Current source (IDAC) setting: Controls current for output current source on IDAC1 and IDAC2. This is disabled.
Configuration Register 3 (address = 03h, default = 00h)			
7:5	MUX[2:0]	000b	IDAC1 routing configuration: IDAC1 disabled
4:2	MUX[2:0]	000b	IDAC2 routing configuration: IDAC2 disabled
1:0	RESERVED	00b	Reserved (read-only): Always write 0.

Table 6.5: Description of the configuration registers and their applied bit values [41]

The I^{2}C communication on the Arduino is implemented with the two function bool i2cWrite(byte address, byte reg, int len, byte *value) for the write and bool i2cRead(byte address, byte reg, int len, byte *value) for the read operation, where address is the I^{2}C address, reg the register address or command (cf. Table 6.3), value the write/read byte array and len the length of this sequence. These functions are the abstraction layer for the I^{2}C communication using the Arduino library Wire.h. Their return value is the communication status (bool: true for success). The following ADC functions write the configuration, trigger the conversion or read out the channel result.

bool initADC():
For the initialization this is forcing a reset and writes the mentioned configuration to the registers. Between this two events a delay time of 1 ms is considered. For a successful communication the return value is true.

bool startADConversion():
To start or restart a new conversation of the current selected channel, this function is used. It returns true, if the communication worked. The conversion time specified in the datasheet is 0.51 ms for the continuous conversion mode and has to be considered before a read-out event.

bool setChannelGain(int channel, int gain):
This function is setting the ADC channel and gain. The parameter channel is limited to the channels 0 to 2 (AIN0, AIN1, AIN2) and the gain to the values 1, 2 and 4, which is obtained by a switched-capacitor circuit in the device. The PGA is always bypassed. Additionally, a new conversion with the current settings is triggered by the use of the function bool startADConversion(). If the input parameters exceed the limited value or if the communication failed, the function returns false.

float readVoltage():
A new ADC measurement result is acquired by float result = readVoltage(). The function itself reads in the binary ADC value, calculates the the corresponding voltage in V and corrects this with the previously applied gain. In case of a failed read operation the return value is zero.

6.3 Microcontroller Power-Up

This section gives a short summary how the Arduino is set up on power-up or reboot. At that time the laser driver and lock-in amplifier circuit should already be properly connected. The pin connections are denoted at the beginning of the source code files. The initial setup starts with the function void setup() in the Arduino code (Appendix D), which starts the serial communication with a baud rate of 115 200 bit/s, initializes a timer for the measurement routine with an interval of 1 ms and calls the function void setupHardware() of the hardware library (Appendix C). This launches the I^{2}C bus, sets the digital SPI pins and initializes the DDSs as well as the ADC and sets the current for the output stages to zero.

6.4 Measurement Routines and Mode Commands

One global timer routine controls the measurement for different modes. This allows the microcontroller to handle the SPI and I^2C communications in the meantime and still achieve a time-critical measurement considering also the delay times of the DDS, the settling time of the measurement signal and the conversion time of the ADC. For each mode, this routine, containing the reconfiguration of the DDS, the ADC measurement and the serial communication, is operated slightly different. The measurement routine is started every millisecond using the measurement timer (timerID_meas) and the measurement routine event void measurementEvent() (cf. Arduino code, Appendix D).

6.4.1 Timer Routine

The basic structure of the timer routine is presented in Table 6.6 starting with the first cycle at time step 0 ms, going to time step 9 ms and repeating itself afterwards, which leads to a sampling result every 10 ms. For a measurement, where one sweep step (e.g. frequency or phase) is measured several times, the action of time step 0 ms is skipped until the next sweep step occurs.

Time Step	Action Description
0 ms	Set new parameters for both DDS Output stages are disabled during reconfiguration
	wait 7 ms for the LIA output signal to settle
7 ms	Start ADC conversion on channel AIN0
	wait 1 ms during conversion
8 ms	Read out ADC channel AIN0 Start ADC conversion on channel AIN1
	wait 1 ms during conversion
9 ms	Read out ADC channel AIN1 Serial communication of the results to the PC (MATLAB)

Table 6.6: Description of the repeating segment of the timer routine

The results of the measurement are sent over the serial COM port in the order of: Sweep parameter (e.g. frequency or phase), time stamp, voltage of the in-phase channel and voltage of the quadrature channel, separated with a semicolon as delimiter.

6.4.2 Mode Description, Parameters, Serial Commands

There are five measurement modes available in the current Arduino firmware, that can be accessed over serial communication (COM port). The initial commands for these are a specific sequence (ASCII standard), which defines the mode and provides the requested parameters. Each of the modes is considering all devices of the QEPAS setup to be operated, the DDSs, the laser current output stage as well as the dual channel ADC. The modes and their access sequence are described below:

- **Frequency sweep:**
  The frequency of both DDS is swept from the **start** to the **stop** parameter with the given **step** size. Therefore the frequency values (float) can be set in terms of **kHz** or **Hz**. For multiple measurement at one frequency the **points** parameter gives the repetitions and the **current** value sets the laser driver 1 current. For this mode the DDSs are working in quadrature mode. The command sequence is specified like this:
  "MODE SWEEP -f <**start**> <**stop**> <**step**> **kHz** <**points**>x <**current**>mA"

- **Phase sweep:**
  For a given **frequency** the phase of the second DDS (quadrature) changes in respect to the first DDS from the **start** to **stop** phase with a resolution of **step** in °. The **points** per step and the laser **current** are set as before.
  "MODE SWEEP -p <**frequency**> **kHz** <**start**> <**stop**> <**step**> degree <**points**>x <**current**>mA"

- **Single frequency mode:**
  This single frequency mode starts a measurement at the specified **frequency** and measures the ADC values for either several times (**points**) or a given **time** period in minutes, conditioned by the different syntax.
  "MODE SINGLE -f <**frequency**> **kHz** <**points**>x <**current**>mA"
  "MODE SINGLE -f <**frequency**> **kHz** <**time**>min <**current**>mA"

- **Locked frequency sweep:**
  In the locked-frequency mode the first DDS is set to the fixed (**locked**) frequency, whereas the second DDS varies its frequency from **start** to **stop** in linear steps of **step**. This function is useful to self-test the dual lock-in amplifier as it was done for the frequency response measurements in Section 5.4.5.

”MODE LOCKEDSWEEP -f <**locked**> <**start**> <**stop**> <**step**> **kHz** <**points**>x <**current**>mA”

- **Decadic locked frequency sweep:**
  This mode is similar to the previous mode, but steps the frequency value in decades. If the parameter **decade** = 1, the measurement begins at the **start** frequency and increases by one decade every step until **stop** is reached. If this value is unequal 1, the step size is in decades (e.g. decade = 0.1 → 10 steps/decade).
  ”MODE LOCKEDSWEEP -d <**locked**> <**start**> <**stop**> <**decade**> **kHz** <**points**>x <**current**>mA”

Additional to the listed commands there are two commands without parameters. One is the stop measurement command, which cancels the running measurement by sending ”STOP”. The second one is ”RESET”, that reboots the whole microcontroller and reinitializes the connected circuitry at the boot-up. The the end of a command is determined with the newline operator ”\n”. All of the commands are not case-sensitive, the decimal separator for floating point number notation however has to be a dot. If a command is not accepted by the parsing function on the Arduino, a serial response will return ”Command not valid!”. The end of each measurement is denoted by the return sequence ”measurement done!”.

6.5 Serial Communication and Command Parsing

The serial communication and the parsing function is realized in the Arduino code (Appendix D). Whenever a character or sequence is sent to the Arduino over the serial communication, the function void serialEvent() is called. This char-wise collects the command up to 200 characters until the newline determination ”\n” occurs. The available command is then analysed by the function bool parseCommand(). If the syntax can not be successfully identified, the buffer is cleared and the return string ”Command not valid!” is sent back. Else, the command is analysed according to the available measurement modes, the required parameters read out and the global variables for the measurement set. Before the timed measurement routine void measurementEvent() is started, a sequence with information about the initiated measurement is returned, e.g. ”Begin frequency sweep: 32000; 33000; 1 Hz; 10x @ 50mA”, and a description of the following data and units, e.g. ”frequency [Hz]; time [ms]; V_inphase [V]; V_quadrature [V]”.

6.6 MATLAB Communication Script

On the PC side, a MATLAB script handles the communication to the Arduino microcontroller, receives the result data and processes them. The general structure of the MATLAB code is divided in six sections: The definition of the measurement parameters, the Arduino connect and read-in, the Arduino shutdown, the save data section and the time-based plot as well as the averaging and plotting. The idea of this concept is to only adapt the measurement parameters for a measurement, run the script and receive the results and plots. The complete code can be found in Appendix E for detailed information.

In the measurement parameter section the COM port of the Arduino, which is displayed in the device manager on Windows, the measurement mode and the parameters are defined. Furthermore, the corresponding command sequence is created. The implemented measurement modes and the corresponding input parameters are briefly listed in the following:

- **Frequency sweep** (mode = 1): The frequency sweep is performed from **freq_start** to **freq_stop** in steps of **freq_step** with the laser current **current** and a repetition per step of **points**.
- **Phase sweep** (mode = 2): The phase is swept from **phase_start** to **phase_stop** in steps of **phase_step** at a frequency of **freq_single** with the laser current **current** and a repetition per step of **points**.
- **Single frequency, n-times** (mode = 3): A measurement at a single frequency of **freq_single** is performed with the laser current **current** and a repetition of **points**.
- **Single frequency, in minutes** (mode = 4): The single frequency measurement is performed for a time in minutes (**min**).
- **Locked frequency sweep** (mode = 5): A locked frequency sweep from **phase_start** to **phase_stop** in steps of **phase_step** is started with **freq_locked** as fixed frequency, the laser current **current** and a repetition per step of **points**.
- **Decadic locked frequency sweep** (mode = 6): The same as locked frequency sweep with a decadic step size.

The Arduino connect and read-in section opens the serial COM port with a baud rate of 115 200 bit/s, sends the measurement command and continuously reads in the data from the Arduino until the end command "measurement done!\n" is received. As long as the measurement is running, this loop is running and MATLAB is busy. To abort this, the key combination Ctrl + C can be used. In this case the Arduino shutdown section has to be executed in order to close the serial communication properly.

After a successful measurement the results are saved in a .txt file in the next code part, followed by the plot sections, which are depending on the measurement mode perform the required calculation (e.g. averaging, vector computation) and represent the data graphically.

CHAPTER 7

Results and Discussion

The final measurements of this thesis are presented in the following chapter. In the introduction, the quartz-enhanced photoacoustic spectroscopy (QEPAS) setup is reviewed and a further description of how the measurements are performed is given. As a performance comparison of the developed, customized lock-in amplifier (LIA) to the former configuration with National Instruments PXI measurement unit, a comparison is provided with an external signal generator for the frequency sweep (chirp). Furthermore, the measurement results of the standalone quartz tuning fork (QTF) without nitrogen dioxide (NO_2) gas flow and the NO_2 concentration measurement are presented.

7.1 Introduction

Apart from the comparison measurement in Section 7.2, the measurements are performed in the measuring configuration consisting of the focused laser diode, the measurement (gas) chamber with the QTF and the surrounding circuitry of laser driver, the lock-in amplifier and the microcontroller (cf. Chapter 3). For the nitrogen dioxide concentration measurements, the NO_2 gas is applied into the measurement chamber with a constant gas flow in different concentrations. This variable concentration is achieved by a gas diluter, which dilutes a 20 ppm NO_2 and synthetic air gas mixture with pure synthetic air to lower NO_2 concentration [46]. The gas diluter with the different mixing ratios, indicated by the letters **A, B, C,...** to **K**, and the synthetic air gas supply (upper pipe) and NO_2 supply (lower pipe) is shown in Figure 7.1.

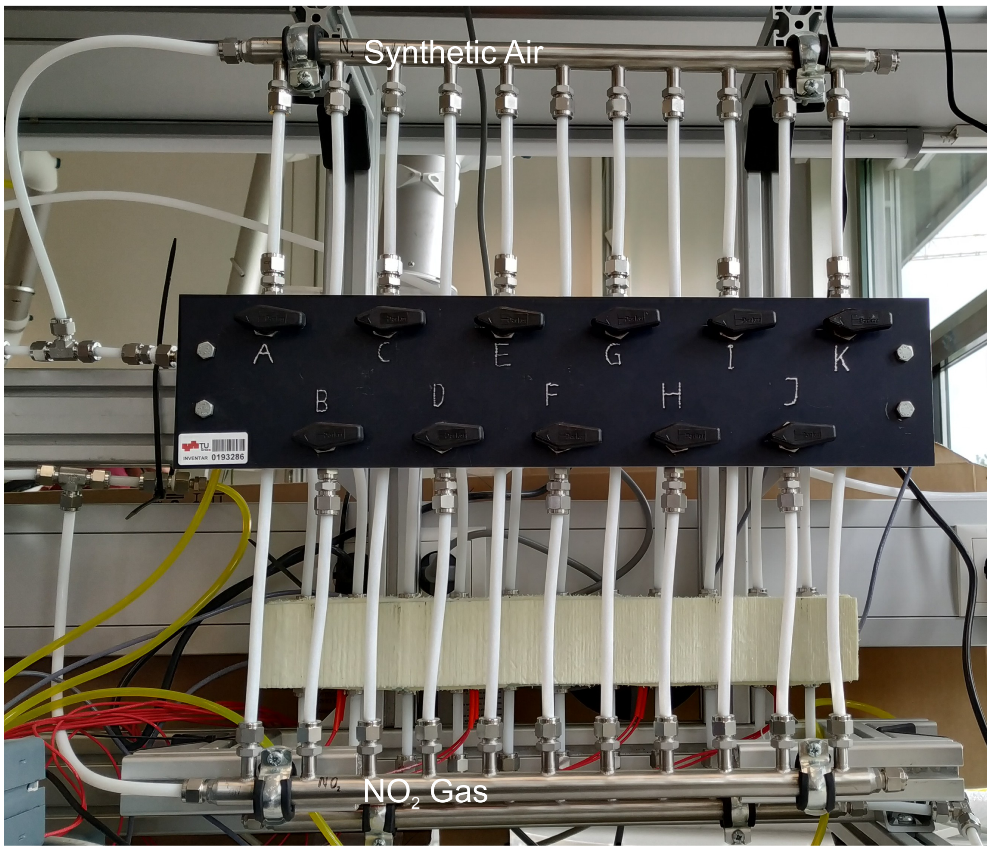


Figure 7.1: Gas diluter with the gas dilution ratios A, B, C,... to K and the gas supply pipes for synthetic air (upper pipe) and NO_2 (lower pipe)

The optical measurement setup is presented in Figure 7.2. The laser beam is collimated by lens 1 and the aperture before being focused into the measurement chamber by lens 2, which is electrically shielded with aluminium foil and copper tape. Through the gas inlet the measurement gas is inserted into the chamber and exhausted by the gas outlet. For the exact positioning in the x- and y-axis, the chamber is mounted on a positioning translation stage.

In Figure 7.3, the controlling circuitry with its labeled signal paths is pictured. The Arduino is connected to the PC over the COM port and controls the laser driver and lock-in amplifier circuit circuits. The laser driver generates the modulation frequency for the laser current output and in addition the sine and cosine reference signal for the lock-in amplifier. This is amplifying the QTF input signal and transmits the result data to the Arduino. The dimensions of the circuitry are relatively small compared to the PXI system and fit well in a PCB housing with 100 x 90 x 50 mm (L x W x H).

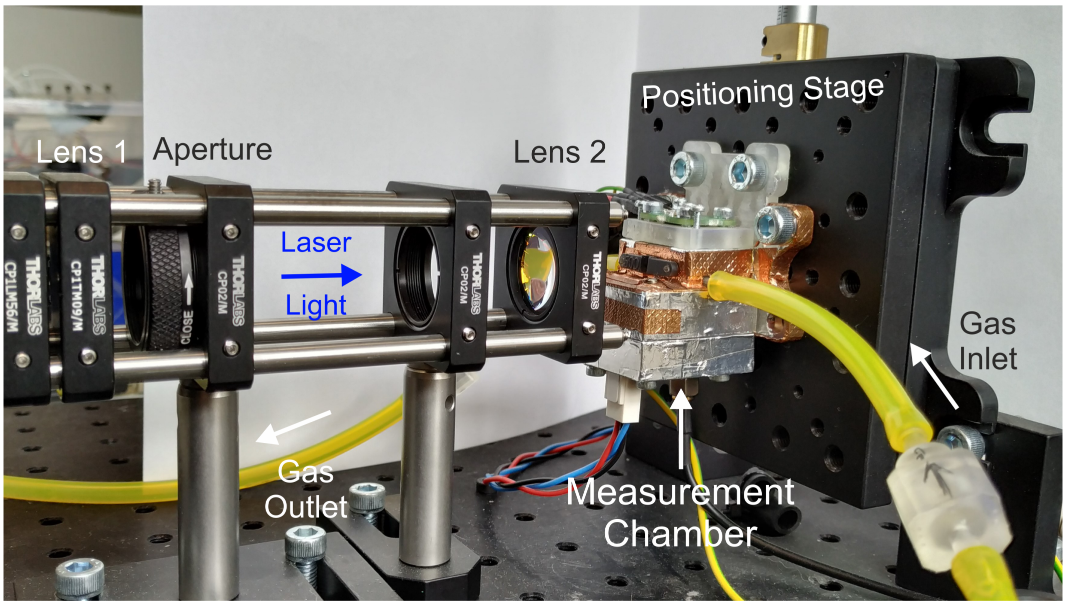


Figure 7.2: Optical arrangement of the measurement setup with lens 1, the aperture and lens 2 for the focusing of the laser beam into the shielded measurement chamber, which is mounted on a positioning stage and supplied with the measurement gas through the gas in- and outlet

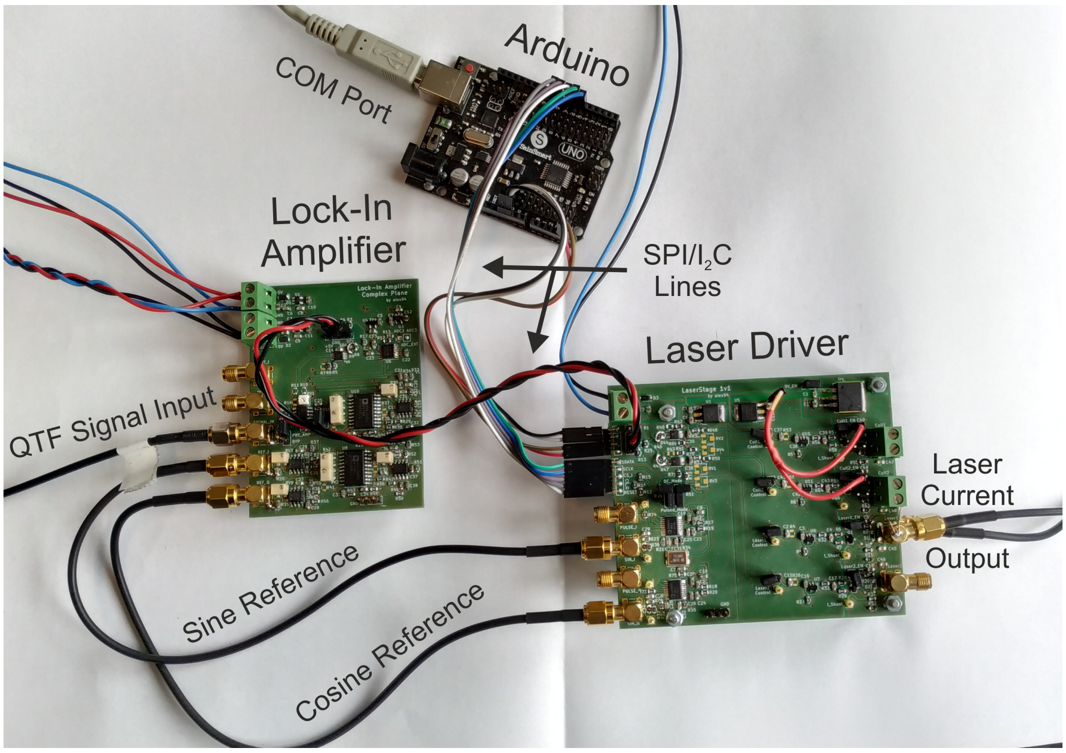


Figure 7.3: Circuitry consisting of the Arduino microcontroller, laser driver and lock-in amplifier and the associated signal paths

7.2 Lock-in Amplifier Comparison

The comparison measurement of the custom-build LIA to the software LIA using NI's PXI units is done simultaneously with an external signal generator, which performs a relatively slow frequency chirp from 32.74 kHz to 32.78 kHz, and a fixed frequency generated by the DDS of the laser driver with an amplitude of 300 mV. The second DDS is intentionally not used for the frequency sweep in order to have an independent phase relation of both signals. The software LIA is operated at an integration time of 1 s and is compared to the custom LIA with a comparable integration of 1 s and 2 s. These integration times result from the ADC sampling rate of 100 SPS and an averaging of 100 Samples for 1 s respectively an averaging integration over 200 Samples for a 2 s integration time. Figure 7.4 shows the both frequency responses graphically.

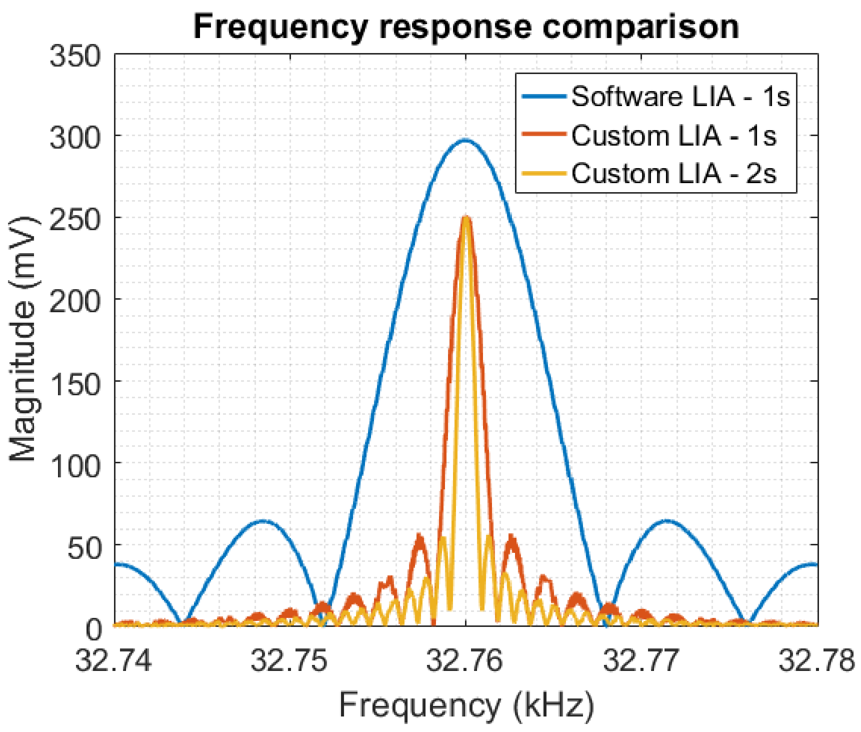


Figure 7.4: Comparison of software LIA with 1 s integration time (blue) to the custom-build LIA with 1 s (red) and 2 s (yellow) integration

The narrow bandwidth of the custom LIA of 2 Hz full width at half maximum (FWHM) in contrast to the ∼10 Hz FWHM of the software LIA for equal integration times is a increase of narrowness by a factor of 5. A larger integration time of 2 s reduces the FWHM bandwidth to 1 Hz. The decreased peak amplitude of 250 mV (custom LIA) is caused by the losses at the analog low-pass filtering as discussed in Section 5.4.3.

7.3 Quartz Tuning Fork Measurements

The standard quartz tuning fork (QTF), that is also found in quartz watches, has a nominal resonance frequency of 32.768 kHz in its sealed condition. However, after removing the cap of the QTF and depending on environmental conditions such as temperature and pressure, the frequency deviates lightly from this nominal frequency. This is why the resonance frequency has to be determined before the actual measurement by a frequency sweep in the most likely range of 32.768 $\pm$0.1 kHz. A second interesting measure regarding the QTF is the long-term signal stability to evaluate influences of drift due to heating by its own vibration. Both results are discussed in the following.

7.3.1 Determination of the Resonance Frequency

For the resonance frequency determination of the QTF, a sweep of the laser modulation frequency is performed and the QTF output signal is detected by the lock-in amplifier (LIA) with the modulation frequency and its quadrature signal (90° phase-shifted) as reference signals. If the laser modulation frequency equal to the QTF's resonance frequency, the amplitude of its output is at the maximum. In Figure 7.5, this frequency sweep is shown in the range from 32.79 kHz to 32.84 kHz with an step size of 0.5 Hz. The resulting magnitude and phase are calculated from the in-phase and quadrature signal after lock-in amplification, measured by the ADC, according to Equation 2.19. The representative integration time of 1 s is accomplished by the 100 SPS sampling rate and an averaging of 100 Samples. For this specific QTF, a resonance frequency of 32.8155 kHz at a phase of 240.32° regarding to the in-phase reference (modulation signal). The FWHM bandwidth of the QTF resonance is measured with 5 Hz. The Q-factor of $Q = 10\,939$ results from the squared magnitude plot by applying Equation 2.12, which is comparable with the numbers from literature around $Q \approx 8000$ to $10\,000$. Due to the restricted domain of the arctangent between $+\frac{\pi}{2}$ and $-\frac{\pi}{2}$, the angle of phase for these measurements is redefined to the range of 0° to 360° in order to avoid a phase jump. The angle Θ is obtained by the in-phase signal X and the quadrature signal Y as follows:

$$\Theta = \begin{cases} arctan\left(\frac{Y}{X}\right) \cdot \frac{360^\circ}{2\pi}, & for\ Y > 0 \\ arctan\left(\frac{Y}{X}\right) \cdot \frac{360^\circ}{2\pi} + 360^\circ, & for\ Y < 0 \end{cases}$$

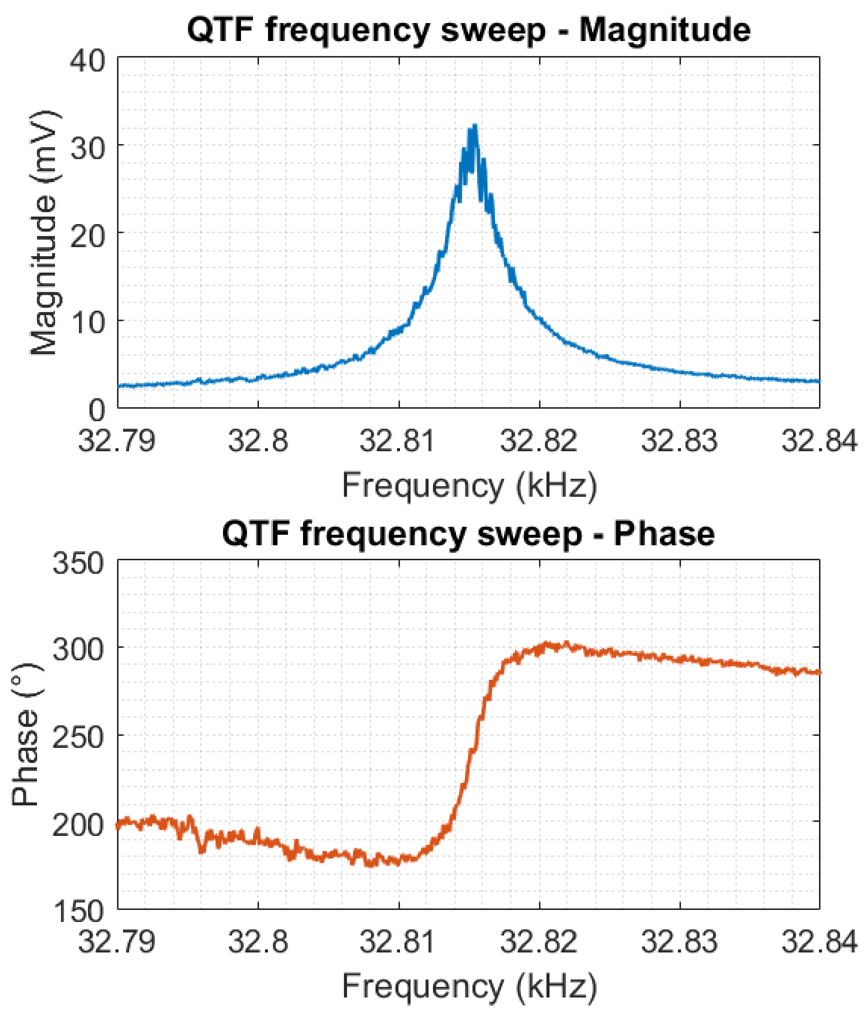


Figure 7.5: Frequency sweep of the laser modulation from 32.79 kHz to 32.84 kHz in 0.5 Hz steps for the determination of the resonance frequency of 32.8155 kHz at a respectively phase of 240.32° to the modulation signal

7.3.2 QTF Signal Stability and Deviation

The QTF signal stability is obtained by a single frequency scan at the specific resonance of 32.8155 kHz over a period of 10 min with the same sampling settings as before. The calculated magnitude and phase is plotted in Figure 7.6 with the corresponding average values and standard deviations. For the magnitude an average value of 33.01 mV at a standard deviation of 1.10 mV is acquired and for the resulting phase an average of 249.11° at a deviation of 1.72°. With this regards, it should be noted that the standard deviation corresponds to the already calculated units and not to the measured signals from the in-phase and quadrature channels. These channels from the two LIA paths have practically the same standard deviation. The conclusion out of this measurement is, that the QTF has a relatively good stability for constant environmental conditions and no recognisable drift, caused by the vibration, occurs.

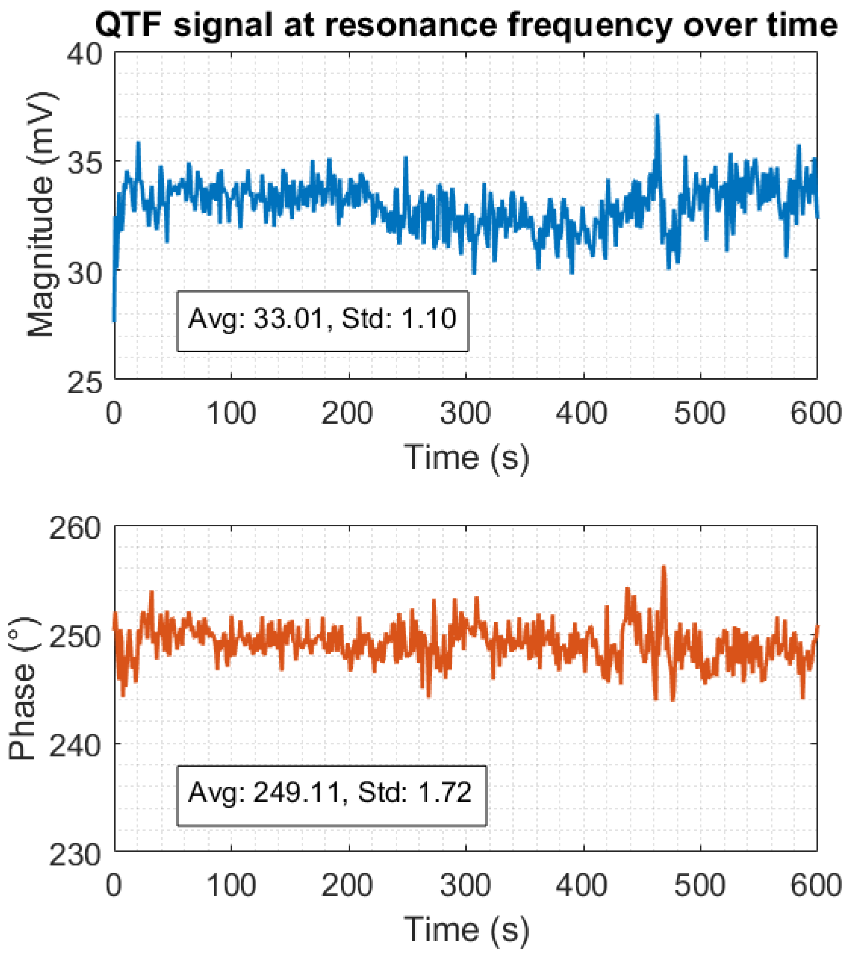


Figure 7.6: Long-term stability of the QTF signal in terms of magnitude and phase over a period of 10 min including the corresponding average values and standard deviation

7.4 Nitrogen Dioxide Concentration Measurements

On the same setup with the 32.8115 kHz QTF the nitrogen dioxide (NO_2) measurements are performed by applying different gas concentrations to the measurement chamber. The gas flow of the diluted NO_2-synthetic air mixture is set by a mass flow controller to 0.3 l/ min. The gas inserts the measurement chamber through the gas inlet and its gas outlet is connected to an exhaust to avoid higher concentrations of the toxic gas in the laboratory area. The measurement with a stepwise increase of the NO_2 concentration and the determination of the detection limit is presented in the next two sections.

7.4.1 NO_2 Concentration Steps

For the NO_2 concentration measurements, a continuous data acquisition during the increase of concentration from 0 ppm to around 20 ppm NO_2 is carried out with a dwell time of approximately 5 min per step. The exact concentrations for the dilution ratios are noted in the measurement plot. Since the QTF already responses to the laser modulation without any photoacoustically excited NO_2 molecules, this averaged background signal, which is determined at a 0 ppm NO_2 concentration or pure synthetic air, is subtracted from the in-phase as well as the quadrature channel voltages before further calculations follow. This measurement is done with an larger integration time of 10 s by averaging 1000 Samples at a sampling rate of 100 SPS. The result of the seven measured concentrations over a time period of 35 min is shown in Figure 7.7 indicating the continuous QFT measurement signal in gray and the steady concentrations steps in color.

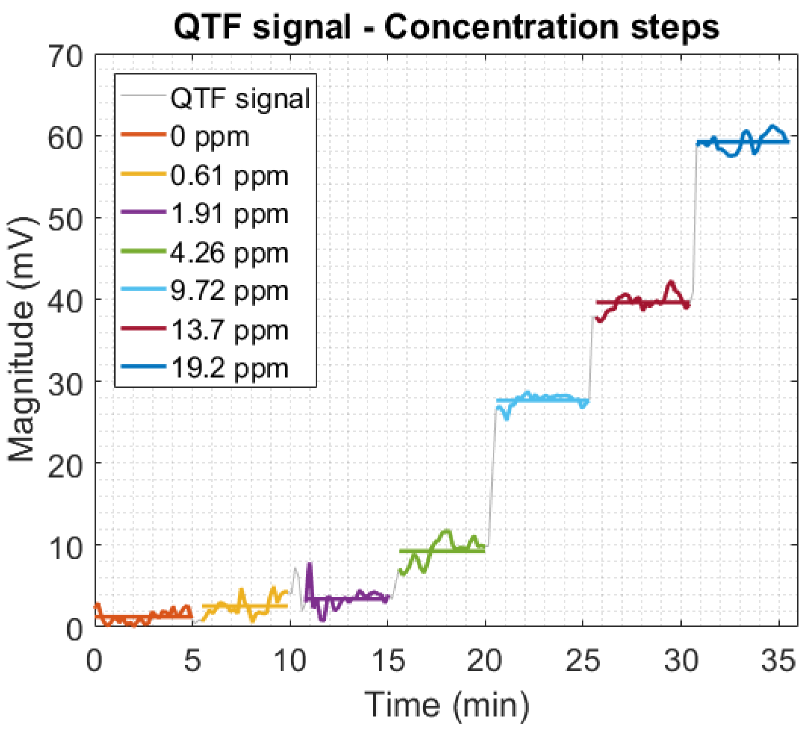


Figure 7.7: Concentration steps of NO_2 from 0 ppm to ~20 ppm at a dwell time of 5 min per step. The gray trace results from the continuous measurement and the colored sections correspond to the steady concentration steps with their indicated mean value.

7.4.2 Limit of Detection

Based on the NO_2 concentration measurement, the QTF signal amplitude is fitted to concentrations between 0 ppm and 20 ppm using a linear fit as presented in Figure 7.8. The error bars show the standard deviation for each measurement point.

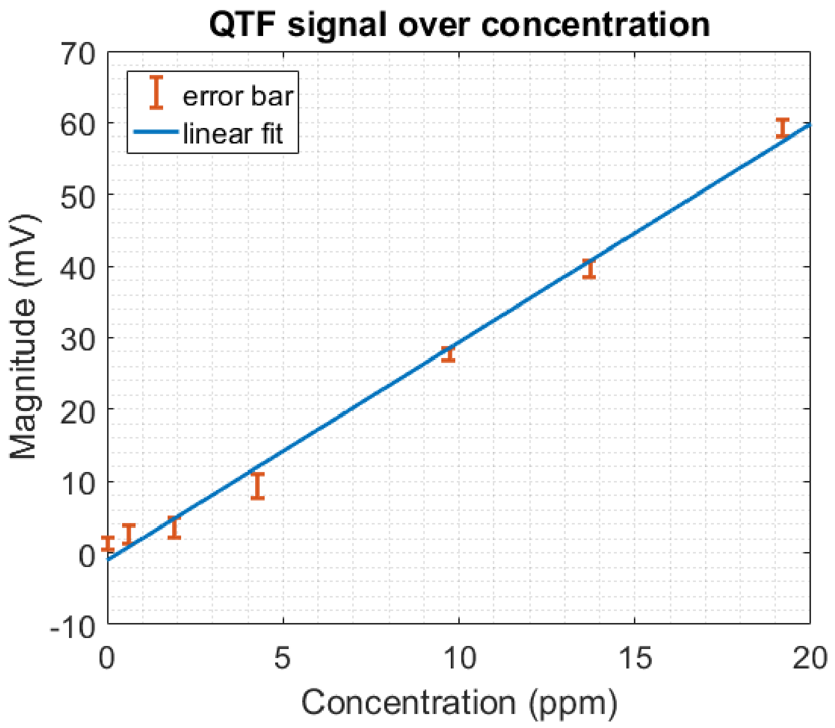


Figure 7.8: Linear fit of the QTF signal amplitude over concentration with error bars representing the standard deviation of the measurement point

For this linear concentration curve, the polynomial description of the signal magnitude is $y = b \cdot x + a$, where $b = 3.0419\,\mathrm{mV/ppm}$ is the slope of the linear regression, $a = -1.0376\,\mathrm{mV}$ the zero-crossing and x the concentration. To derive the 3-sigma limit of detection (LOD), following equation for the linear regression is applied with S_0 as the standard deviation of the blank sample at 0 ppm [47].

$$LOD = 3 \cdot \frac{S_0}{b} = 0.7842\,\mathrm{ppm} = 784.2\,\mathrm{ppb}$$

In comparison to the measurement results on the existing system with the high cost and bulky PXI measurement unit, which came to a calculated 3-sigma LOD of 200 ppb, this measurement showed a degradation by a factor of around 4 with the same integration time of 10 s on the PXI system [1].

7.5 Discussion

Retrieving the results from this chapter, the measurement system with the custom developed circuitry is capable of detecting NO_2 concentrations in the sub-ppm range. The lock-in amplification using the dual-phase LIA in contrast to the available system performs better regarding the narrowness of bandwidth and roll-off at the locked frequency. It is furthermore possible to determine the resonance frequency of the QTF to a precision of 0.5 Hz. For the signal stability of the QTF a major drift over a longer period could be excluded and a deviation of around 1.1 mV for both LIA output channels stated. Somehow, the NO_2 concentration measurement is not performing as well and the concentration steps show noise ripples in the millivolt range. On the one hand this is caused by the variance in the signal stability, on the other hand a additional impact is coming from the gas flow, which increases the noise level of the QTF signal. Remembering the pass-through of the odd harmonics in the LIA evaluation, due to the quasi-multiplication of the signal and a square wave, the noise density from these harmonics is added to the total noise (cf. Section 5.4.5 and Figure 5.16). By adding a band-pass filter or possibly only a low-pass filter of higher order in front of the LIA signal input to cut off the harmonics and filter only the frequency of interest, the signal-to-noise ratio might be tremendously improved. This could lead to a reduced deviation of the signal stability and a lower negative impact of the gas flow, which probably still produces a small broadband background noise, which can propagate in the bands of the odd harmonics.

CHAPTER 8

Conclusion and Outlook

In general, the custom developed circuits work reasonable well in performance. The laser driver delivers a satisfying current pulse-shape at a remarkable frequency accuracy, generated by the direct digital synthesizer (DDS). The synchronisation of the two DDSs is working as expected and enables them to provide the required in-phase (sinus) and quadrature (cosine) reference signals for the dual-phase-sensitive detection. The achieved frequency resolution using the 8 MHz quartz oscillator as the reference clock for the DDSs is 0.03 Hz. From the measurements with the oscilloscope, a constant frequency offset of −0.3 Hz was detected, which is most-likely caused by the oscillator frequency tolerance. Since this is a constant error, it can be compensated by tuning the frequency in software. The sine and cosine reference signal outputs with a sampling rate of around 240 Samples/period at the QTF resonance frequency are perfectly generated and their phase relation of ∼90° proven. Therefore the second of the two DDSs is operated with a 90° phase configuration, which is precise to a calculated 0.09°. The current pulse amplitude can be controlled by the 7-bit digital potentiometer in the range from 0 mA to 100 mA with a step size of 0.78 mA. In the evaluation this was performing well with a small deviation of 2 to 3 mA, which is within the acceptable tolerances. The short overshoot of up to 25 % in the current pulse shape is intentionally tolerated in order to reach faster rise and fall times of less than 0.5 µs. For the laser application, this overshoot has no further disadvantages.

The functionality of the concept with the dual-phase lock-in amplifier (LIA) using a synchronous demodulator as phase-sensitive detector is proven in this work. This circuit includes the pre-amplification for sine and cosine reference as well as QTF input signal, the two identical lock-in amplifiers, followed by a 4^{th} order low-pass filter and the analog-to-digital converter (ADC). Because the reference signals (sine and cosine) are unipolar and have a small, 300 mV amplitude, these two signals are AC coupled and pre-amplified at the input. The 10-fold amplification to a signal amplitude of ± 3 V is working well with adjusted common-mode offset of less than 1 mV for both channels. The pre-amplification of the QTF input signal can be set by a manual potentiometer on the PCB to values between 2 x and 1000 x gain or be bypassed. This is used for an additional amplification for very small inputs, whereas the maximum output amplitude is limited to ± 2.5 V. The synchronous demodulators in LIA configuration operate according to the expectation and simulation results on the basis of a digital switching multiplier. In the latest measurements with the standard settings of 100 SPS and an averaging of 100 Samples, corresponding to an integration time of 1 s, the FWHM bandwidth of the LIAs was determined to 0.6 Hz at the nominal QTF resonance frequency of 32.768 kHz. This is a extremely narrow frequency response with a calculated Q-factor of around 54 600. Based on the general principle of a digital switching multiplier, the impact of higher odd harmonics, that can pass the LIA with an attenuation 1/order of the harmonic, was detected. This is not filtered out with the low-pass filters at the LIA output and add additional noise to the measurement result. The only way to avoid this and cut off the higher harmonics is to add a bandpass or low-pass filter with some orders of attenuation in front of the LIA signal input. For the bipolar ADC, a voltage resolution of 298 nV in the range from +2.5 V to −2.5 V can be achieved and a maximum sampling rate of 2000 SPS, which is limited by the Arduino firmware to 100 SPS.

The cooperation of analog circuitry and the microcontroller as well as the communication with MATLAB is working smoothly on account of the self-developed firmware. The MATLAB script connects to the Arduino microcontroller via the serial COM port over USB and is able to trigger a specific measurement mode with the defined control sequences. For each mode the Arduino handles the whole analog circuitry and measures the ADC results. During the measurement, MATLAB is continuously receiving the data including the time stamp, the sweep parameter (e.g. frequency, phase) and the in-phase and quadrature channel voltages. This raw data is automatically saved in a .txt file. Additionally, some calculations such as averaging and magnitude/phase computation are performed and the preliminary results plotted.

In the measurements on the quartz-enhanced photoacoustic setup, the parameters of the quartz tuning fork (QTF) could be determined and the detection of nitrogen dioxide (NO_2) concentrations using the custom circuits demonstrated. For the determination of the QTF parameters, the setup was operated of ambient conditions without a gas flow. The resonance frequency of the current QTF was determined to 32.8115 kHz with an uncertainty of less than 0.5 Hz. Its FWHM frequency bandwidth was measured with 5 Hz and the Q-factor calculated to $Q = 10\,939$. The stability of the QTF signal was measured over a longer period of time with a standard deviation in the range of 1 mV channel voltage or 1.1 mV respectively 1.72° in terms of magnitude and phase. For the NO_2 concentration measurement, a stepwise increase of NO_2 concentration in a gas mixture of synthetic air is performed from 0 ppm to 19.2 ppm and applied to the measurement chamber. The QTF response at 0 ppm NO_2 in synthetic air is taken as background signal and subtracted from the measurement series. The results from this concentration measurement in general show a linear dependency between magnitude and NO_2 concentration, whereas the concentration steps have noise ripples of some millivolts. By the method of the linear regression, the limit of detection (LOD) is calculated to 784 ppb for an increased integration time of 10 s. This is worse by a factor of 4 compared to the existing PXI reference system with a LOD of 200 ppb. Nevertheless, with further improvements this could lead to a equally good performance and replace the bulky and cost-intensive measurement equipment.

The outlook for this project is to further develop this system including improvements on the circuit and software side, the miniaturization of the optical components to focus the laser beam and the prototyping of a stand-alone measurement device. As mentioned before the bandpass filtering of the QTF signal in the frequency band of interest (around the typical QTF resonance frequency) before being applied to the LIA is one effective improvement. But also investigations on the pre-amplification and noise reductions of the signal on the path from the QTF to the LIA and ADC can help to increase the performance. Furthermore, a lock-in amplifier with an analog multiplier as phase-sensitive detector can be considered or a lock-in amplification in the digital domain using fast analog-to-digital converters is an option. Besides this, there is the possibility to optimize the measurement system by software algorithms such as self-calibration, advanced background signal subtraction and other measures. Either way, it is advisable to switch to a more powerful microcontroller in order to increase the sampling rate to the ADC maximum of 2000 SPS and perform some of the calculations, that are currently done in MATLAB, on the microcontroller. By employing an microcontroller with IoT (Internet of

Things) connection, the measurement system could moreover act as wireless sensor node for air quality monitoring. In this case, a battery-powered application will get difficult as the laser has a high power consumption in measurement mode, but with an external energy supply, this is feasible.

For the miniaturization of the quartz-enhanced photoacoustic setup, the micro-optics of a BluRay drive was considered in the past. The blue/violet laser diode of the drive has a wavelength of 405 nm and is hence not suitable for the NO_2 measurement since photodissociation of the NO_2 molecules already occurs in this range, but the optics in the BluRay module also work for a 450 nm laser diode. The great advantage of this is the cheap, compact and complete optical solution with a small focal point, ideal for the focusing through a quartz tuning fork. Investigations on that have been done, whereat some difficulties arose in the design of the measurement chamber and the positioning of the laser beam due to the short focal length of the exit lens. Further development work on that lead to a extreme down-scaling of the measurement device.

Bibliography

[1] P. Breitegger, B. Schweighofer, H. Wegleiter, and A. Bergmann, "Towards Low-Cost QEPAS Sensors for Nitrogen Dioxide Detection," *Manuscript submitted for publication*, 2019.

[2] J. P. Besson and L. Thévenaz, "Photoacoustic spectroscopy for multi-gas sensing using near infrared lasers," *Laboratoire de nanophotonique et métrologie*, vol. Ph.D., no. Thèse No. 3670 (2006), pp. 1–189, 2006.

[3] A. Miklós and P. Hess, "Peer Reviewed: Modulated and Pulsed Photoacoustics in Trace Gas Analysis," *Analytical Chemistry*, vol. 72, no. 1, pp. 30 A–37 A, 2000. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ac002681m

[4] P. Patimisco, G. Scamarcio, F. K. Tittel, and V. Spagnolo, "Quartz-enhanced photoacoustic spectroscopy: A review," *Sensors (Switzerland)*, vol. 14, no. 4, pp. 6165–6206, 2014.

[5] C. Berger, "Photoakustische Spektroskopie zur Emissionsüberwachung," Ph.D. dissertation, 2017.

[6] W. Murphy, *Modern spectroscopy*, 1993, vol. 4.

[7] I. V. Hertel and C.-P. Schulz, *Atome, Moleküle und optische Physik 2*, 2010. [Online]. Available: http://link.springer.com/10.1007/978-3-642-11973-6

[8] B. Dick, "Praktikum Physikalische Chemie II Absorption und Emission organischer Moleküle Fluoreszenz, Phosphoreszenz, Energieübertragung," p. 4, 2014.

[9] T. Kenny, Martyn; Oates, "Lime and Limestone," *Ullmann's Encyclopeida of industrial chemistry*, pp. Vol. 15 1–40, 1960.

[10] A. D. McNaught and A. Wilkinson, "IUPAC. Compendium of Chemical Terminology,

2nd ed." *http://www.iupac.org/goldbook/A00220.pdf*, 2006.

[11] P. Meyer and M. Sigrist, "Atmospheric-Pollution Monitoring Using CO2- Laser Photoacoustic-Spectroscopy and Other Techniques," 1990.

[12] A. A. Kosterev, Y. A. Bakhirkin, R. F. Curl, and F. K. Tittel, "Quartz-enhanced photoacoustic spectroscopy," *Optics Letters*, vol. 27, no. 21, p. 1902, 2002. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=ol-27-21-1902

[13] Y. L. Liu and R. Zhang, "AD630 Lock-In Amplifier Circuit for Weak Signal," *Advanced Materials Research*, vol. 482-484, pp. 975–980, 2012. [Online]. Available: http://www.scientific.net/AMR.482-484.975

[14] T. Starecki and P. Z. Wieczorek, "A high sensitivity preamplifier for quartz tuning forks in qepas (Quartz enhanced photoacoustic spectroscopy) applications," *Sensors (Switzerland)*, vol. 17, no. 11, 2017.

[15] Zürich Instruments, "Principles of lock-in detection and the state of the art Zurich Instruments," *White Paper*, no. November, pp. 1–10, 2016.

[16] Perkin Elmer Instruments, "Technical Note TN 1000: What is a Lock-in Amplifier?" *Perkin Elmer Technical Notes*, pp. 1–4, 2000. [Online]. Available: http://www.univie.ac.at/photovoltaik/umwelt/ws2015/WhatisaLock-inamplifier.pdf{%}0Ahttp://cpm.uncc.edu/sites/cpm.uncc.edu/files/media/tn1000.pdf

[17] SRS, "About Lock-In Amplifiers," *Application Note*, no. 408, pp. 1–9, 2011.

[18] M. Chiwa, H. Kondo, N. Ebihara, and H. Sakugawa, "Atmospheric concentrations of nitric acid, sulfur dioxide, particulate nitrate and particulate sulfate, and estimation of their dry deposition on the urban- and mountain-facing sides of Mt. Gokurakuji, Western Japan," *Environmental Monitoring and Assessment*, vol. 140, no. 1-3, pp. 349–360, 2008.

[19] G. Lammel and J. N. Cape, "Nitrous acid and nitrite in the atmosphere," *Chemical Society Reviews*, vol. 25, no. 5, p. 361, 1996. [Online]. Available: http://xlink.rsc.org/?DOI=cs9962500361

[20] F. Song, J. Young Shin, R. Jusino-Atresino, and Y. Gao, "Relationships among the springtime ground–level NOx, O3 and NO3 in the vicinity of highways in the US East Coast," *Atmospheric Pollution Research*, vol. 2, no. 3, pp. 374–383, 2011.

[Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S130910421530492X

[21] LUBW, "Zusammensetzung der Luft." [Online]. Available: http://www4.lubw.baden-wuerttemberg.de/servlet/is/18340/ (Accessed 2017-07-05).

[22] H. Keller-Rudek, G. K. Moortgat, R. Sander, and R. Sörensen, "The MPI-Mainz UV/VIS spectral atlas of gaseous molecules of atmospheric interest," pp. 365–373, 2013. [Online]. Available: http://www.earth-syst-sci-data.net/5/365/2013/ (Accessed 2018-11-22).

[23] L. S. Rothman, C. P. Rinsland, A. Goldman, S. T. Massie, D. P. Edwards, J.-M. Flaud, A. Perrin, C. Camy-Peyret, V. Dana, J. Y. Mandin, J. Schroeder, A. McCann, R. R. Gamache, R. B. Wattson, K. Yoshino, K. V. Chance, K. W. Jucks, L. R. Brown, V. Nemtchinov, and P. Varanasi, "Hitran database," pp. 665–710, 1998. [Online]. Available: http://hitran.iao.ru/ (Accessed 2018-11-23).

[24] C. M. Roehl, J. J. Orlando, G. S. Tyndall, R. E. Shetter, G. J. Vázquez, C. A. Cantrell, and J. G. Calvert, "Temperature dependence of the quantum yields for the photolysis of NO2 near the dissociation limit," *Journal of Physical Chemistry*, vol. 98, no. 32, pp. 7837–7843, 1994.

[25] K. Skalska, J. S. Miller, and S. Ledakowicz, "Kinetics of nitric oxide oxidation," *Chemical Papers*, vol. 64, no. 2, pp. 269–272, 2010.

[26] H. Tsukahara, T. Ishida, and M. Mayumi, "BRIEF REVIEW Gas-Phase Oxidation of Nitric Oxide: Chemical Kinetics and Rate Constant," vol. 3, no. 3, pp. 191–198, 1999. [Online]. Available: http://www.idealibrary.com

[27] E. P. Gardner, P. D. Sperry, and J. G. Calvert, "Primary quantum yields of NO2 photodissociation," *Journal of Geophysical Research: Atmospheres*, vol. 92, no. D6, pp. 6642–6652, 1987. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1029/JD092iD06p06642/abstract

[28] W. R. Harshbarger and M. B. Robin, "The Opto-Acoustic Effect: Revival of an Old Technique for Molecular Spectroscopy," *Accounts of Chemical Research*, vol. 6, no. 10, pp. 329–334, 1973.

[29] P. Atkins and J. De Paula, *Atkins' Physical Chemistry 8th edition*, 2006.

[30] Analog Devices, "AD9834 Data Sheet," pp. 1 – 32, 2014. [Online]. Available: http://

www.analog.com/media/en/technical-documentation/data-sheets/AD9834.pdf{%}0Ahttp://www.analog.com/media/en/technical-documentation/data-sheets/AD9102.pdf{%}0Ahttp://www.mouser.com/ds/2/609/AD5626-877177.pdf

[31] Analog Devices, "AD9834 Circuit Note," 2013. [Online]. Available: http://www.analog.com/static/imported-files/circuit{_}notes/CN0156.pdf

[32] Microchip, "Mcp453x/455x/463x/465x Data Sheet," 2013.

[33] Texas Instruments, "LMV793 / LMV794 88 MHz , Low Noise , 1 . 8V CMOS Input , Decompensated Operational Amplifiers," no. March 2007, 2013.

[34] Diodes Incorporate, "DMG2302UK Datasheet," no. January, pp. 1–7, 2016.

[35] Texas Instruments, "LP295x-N Series of Adjustable Micropower Voltage Regulators," 2016.

[36] Texas Instruments, "LM2940x 1-A Low Dropout Regulator," 2014.

[37] Analog Device, "Balanced Modulator / Demodulator AD630 For Lock-In amplifier," 2015.

[38] M. Prevedelli, "Simple lock – in amplifier Rev . 5," pp. 1–12, 2008.

[39] Analog Devices, "Low Noise Precision Operational Amplifier OP27 Datasheet," 2006.

[40] Analog Devices, "Low Cost Low Power Instrumentation Amplifier AD620," 2011.

[41] Texas Instruments, "ADS122C04 24-Bit , 4-Channel , 2-kSPS , Delta-Sigma ADC With I2C Interface," 2018.

[42] Linear Technology, "LT1761 Series - 100mA, Low Noise, LDO Regulators," pp. 1–20, 2015. [Online]. Available: http://cds.linear.com/docs/en/datasheet/1763fh.pdf

[43] Linear Technology, "LT1964 - 200mA, Low Noise, Low Dropout Negative Micropower Regulator," *Complement*, pp. 1–16, 1964.

[44] Texas Instruments, "200mA Low-Noise, High-PSRR Negative Output Low-Dropout Linear Regulators TPS723xx," no. 4, 2003. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps723.pdf

[45] Texas Instruments, "150mA , Ultra-Low Quiescent Current Low-Dropout Linear Regulator TPS782xx," *Exposure*, no. August 2008, 2010.

[46] P. Breitegger and A. Bergmann, "A Precise Gas Dilutor Based on Binary Weighted Critical Flows to Create NO2 Concentrations," *Proceedings*, vol. 2, no. 13, p. 998, 2018.

[47] A. Shrivastava and V. Gupta, "Methods for the determination of limit of detection and limit of quantitation of the analytical methods," *Chronicles of Young Scientists*, vol. 2, no. 1, p. 21, 2011. [Online]. Available: http://www.cysonline.org/text.asp?2011/2/1/21/79345

Appendix

A Laser Driver Schematic

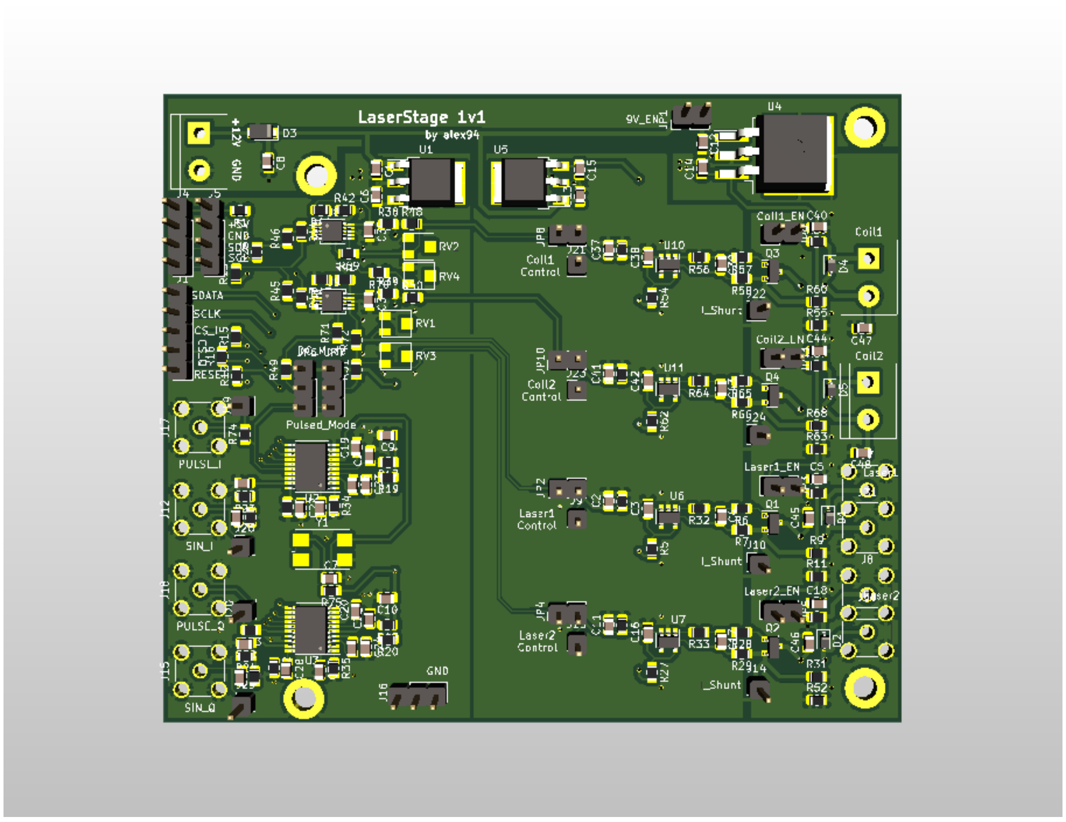
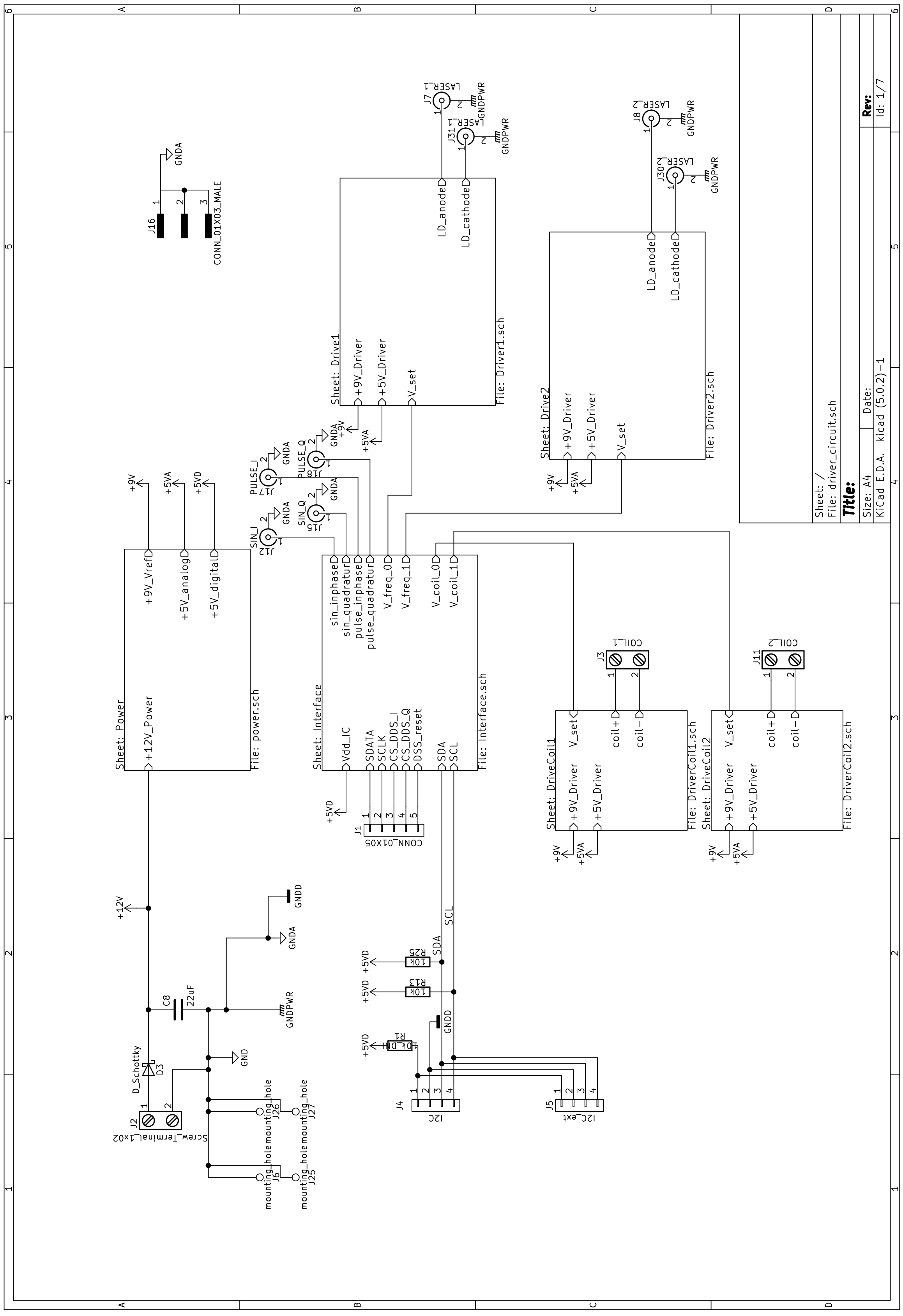
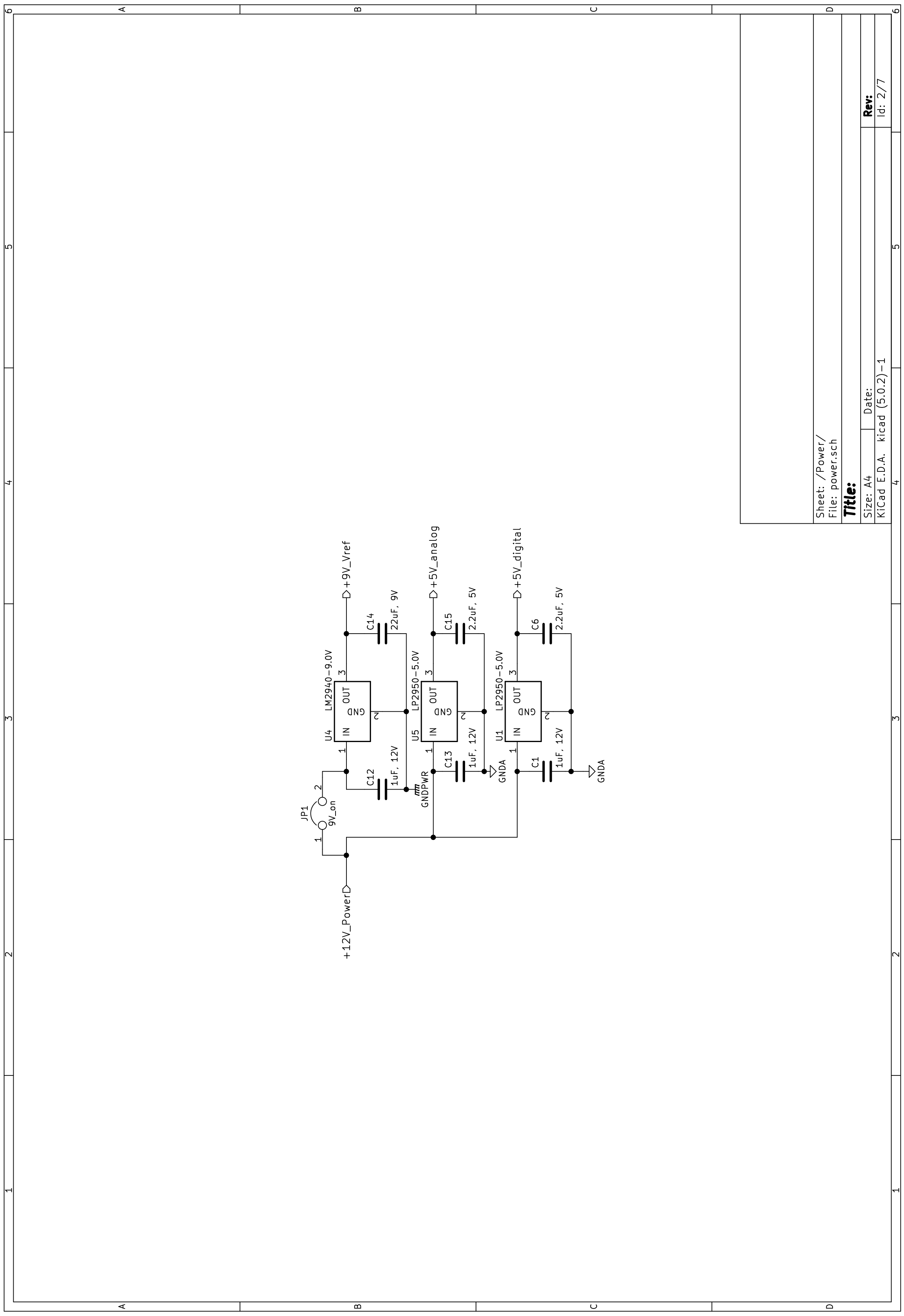
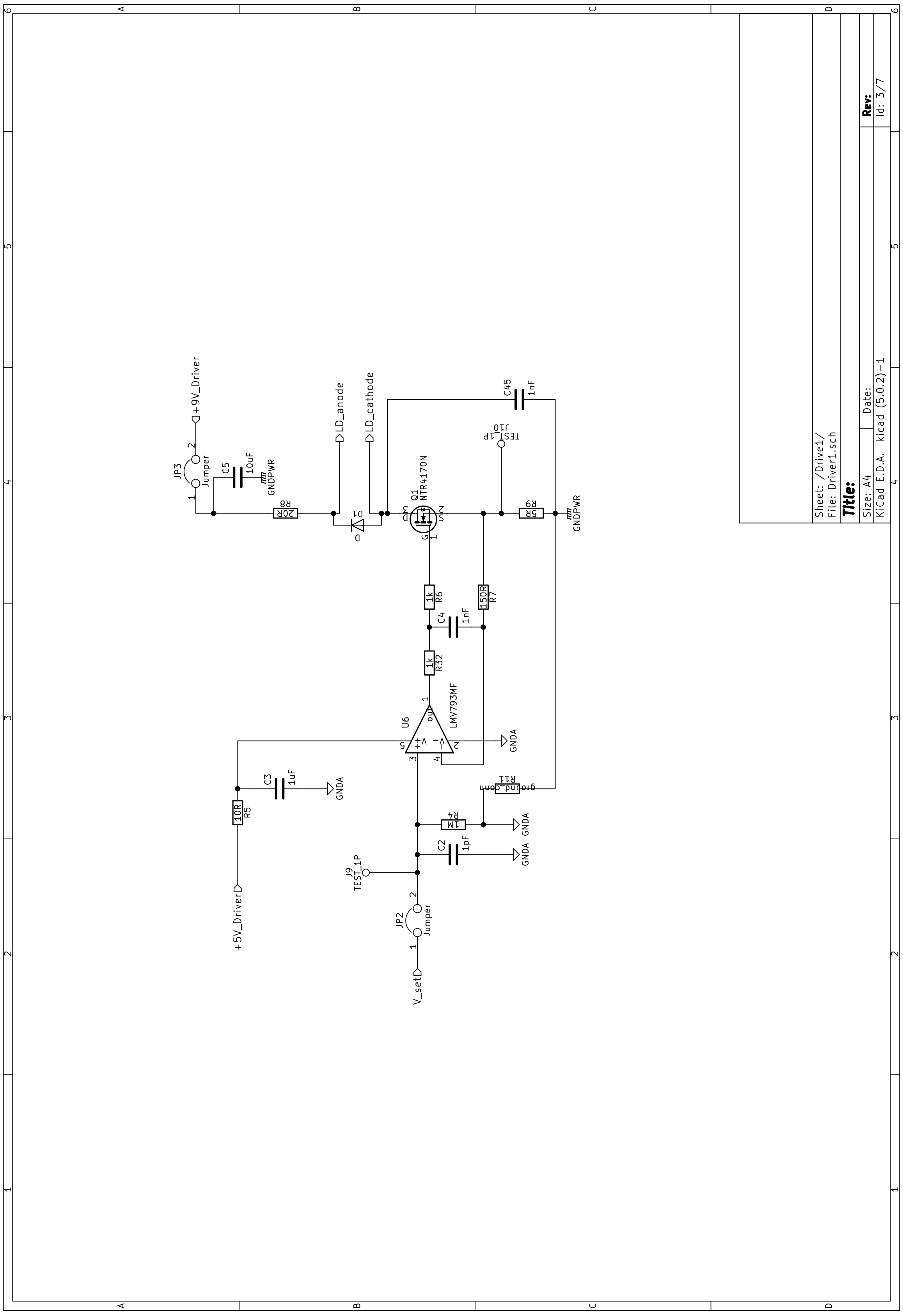
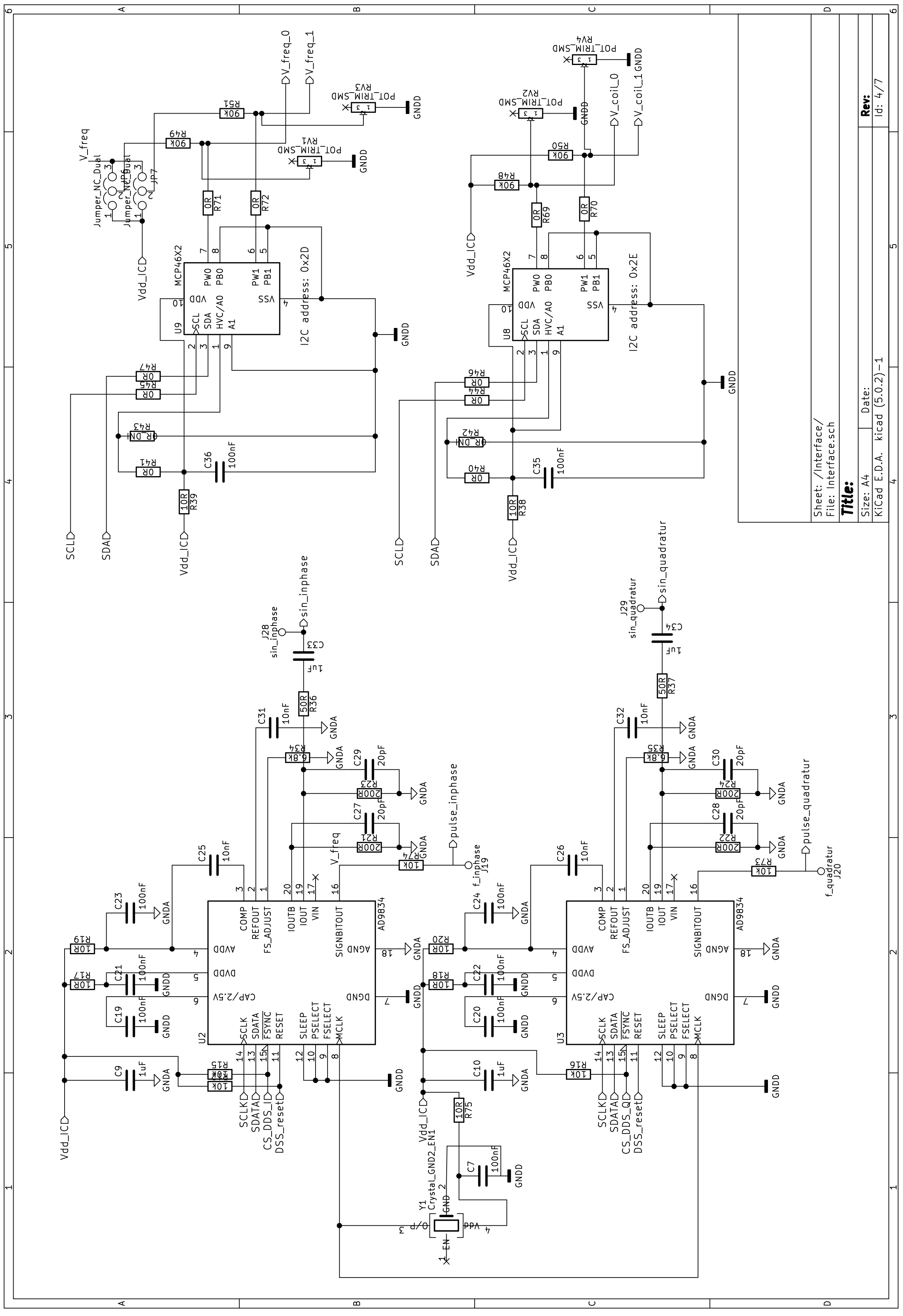


Figure A1: Laser driver schematic in 3D-view

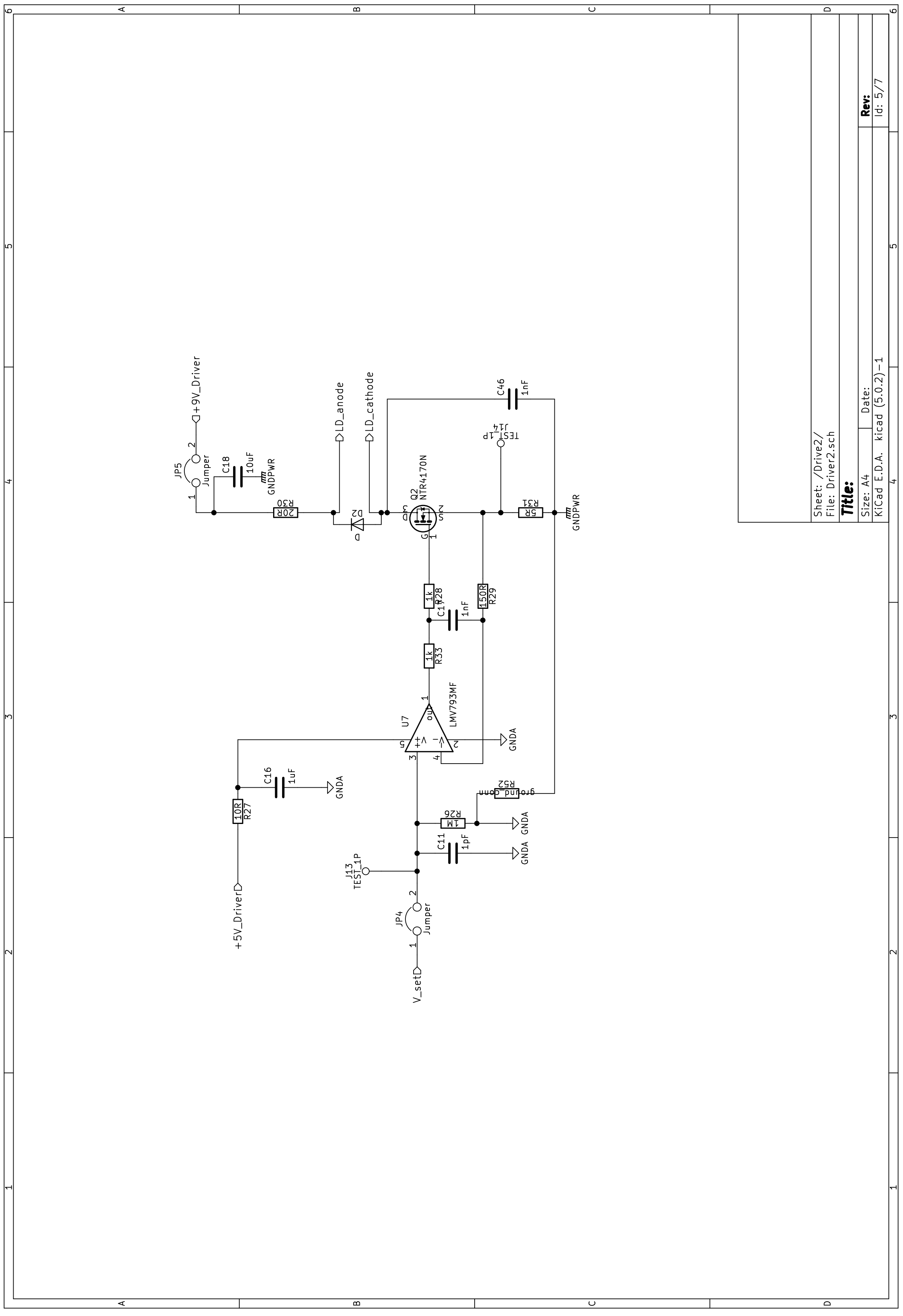

Sheet: Power
File: power.sch
+12V_Power
+9V_Vref
+5V_analog
+5V_digital
Sheet: Interface
File: Interface.sch
sin_inphase
sin_quadratur
pulse_inphase
pulse_quadratur
V_freq_0
V_freq_1
V_coil_0
V_coil_1
Vdd_IC
SDATA
SCLK
CS_DDS_I
CS_DDS_Q
DSS_reset
SDA
SCL
Sheet: Drive1
File: Driver1.sch
+9V_Driver
+5V_Driver
V_set
LD_anode
LD_cathode
Sheet: Drive2
File: Driver2.sch
Sheet: DriveCoil1
File: DriverCoil1.sch
Sheet: DriveCoil2
File: DriverCoil2.sch
coil+
coil−
J3 COIL_1
J11 COIL_2
J7 LASER_1
J31 LASER_1
J8 LASER_2
J30 LASER_2
J12 SIN_I
J15 SIN_Q
J17 PULSE_I
J18 PULSE_Q
J16 CONN_01X03_MALE
J1 CONN_01X05
J4 I2C
J5 I2C_ext
R25 10k
R13 10k
R1 10k
C8 22uF
D3 D_Schottky
J2 Screw_Terminal_1x02
J26 mounting_hole
J27 mounting_hole
J6 mounting_hole
J25 mounting_hole
GNDA
GNDD
GNDPWR
GND
+12V
+9V
+5VA
+5VD
Sheet: /
File: driver_circuit.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 1/7

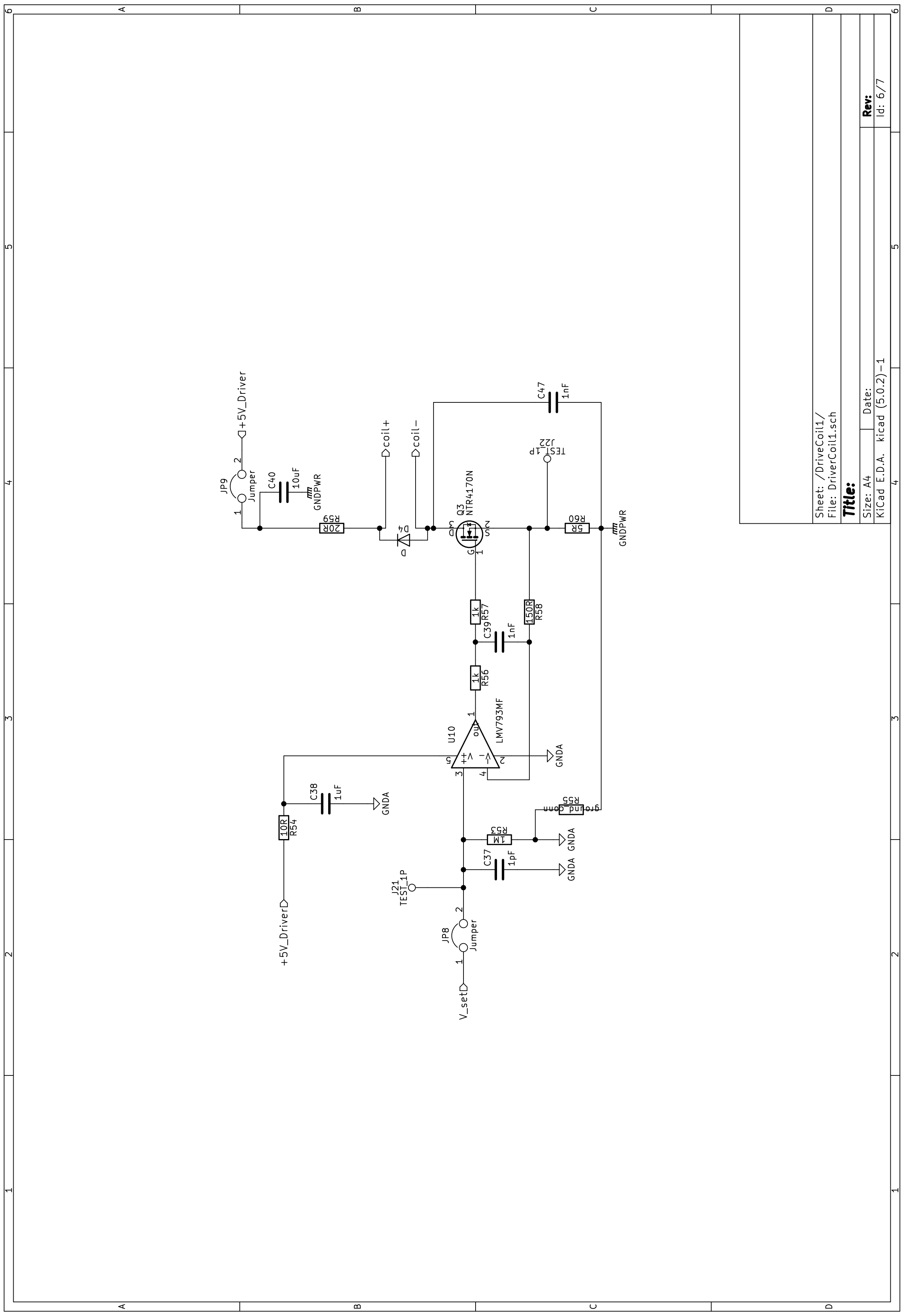

+12V_Power
JP1
9V_on
U4
LM2940-9.0V
IN
OUT
GND
C12
1uF, 12V
C14
22uF, 9V
+9V_Vref
GNDPWR
U5
LP2950-5.0V
C13
1uF, 12V
C15
2.2uF, 5V
+5V_analog
GNDA
U1
LP2950-5.0V
C1
1uF, 12V
C6
2.2uF, 5V
+5V_digital
GNDA
Sheet: /Power/
File: power.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 2/7

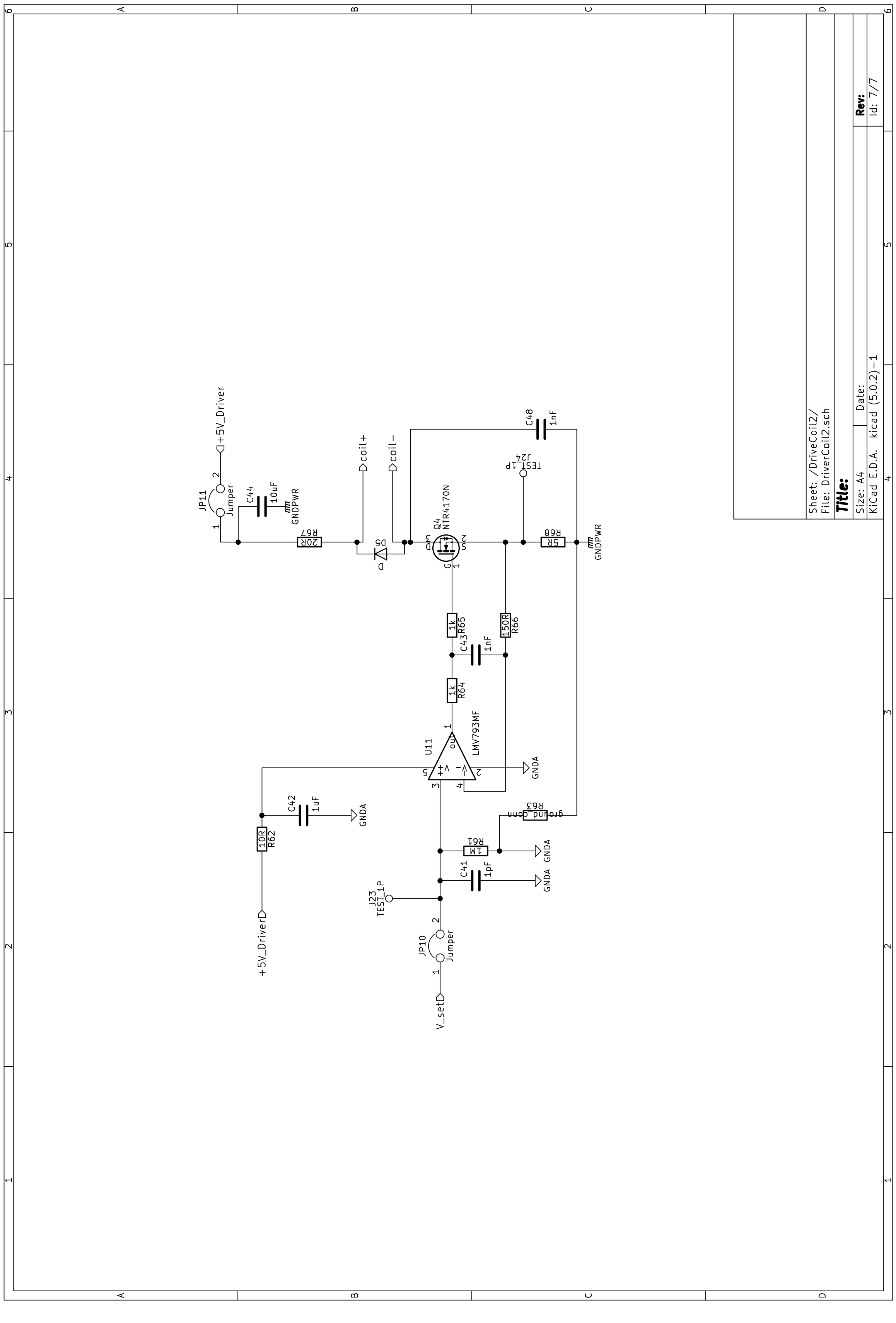

+9V_Driver
JP3
Jumper
C5
10uF
GNDPWR
R8
20R
LD_anode
LD_cathode
D1
D
Q1
NTR4170N
C45
1nF
J10
TEST_1P
R9
5R
GNDPWR
R6
1k
R7
150R
C4
1nF
R32
1k
U6
LMV793MF
GNDA
R11
ground_conn
R5
10R
C3
1uF
GNDA
R4
1M
GNDA
C2
1pF
GNDA
J9
TEST_1P
JP2
Jumper
V_set
+5V_Driver
Sheet: /Drive1/
File: Driver1.sch
Title:
Size: A4
Date:
KiCad E.D.A. kicad (5.0.2)-1
Rev:
Id: 3/7



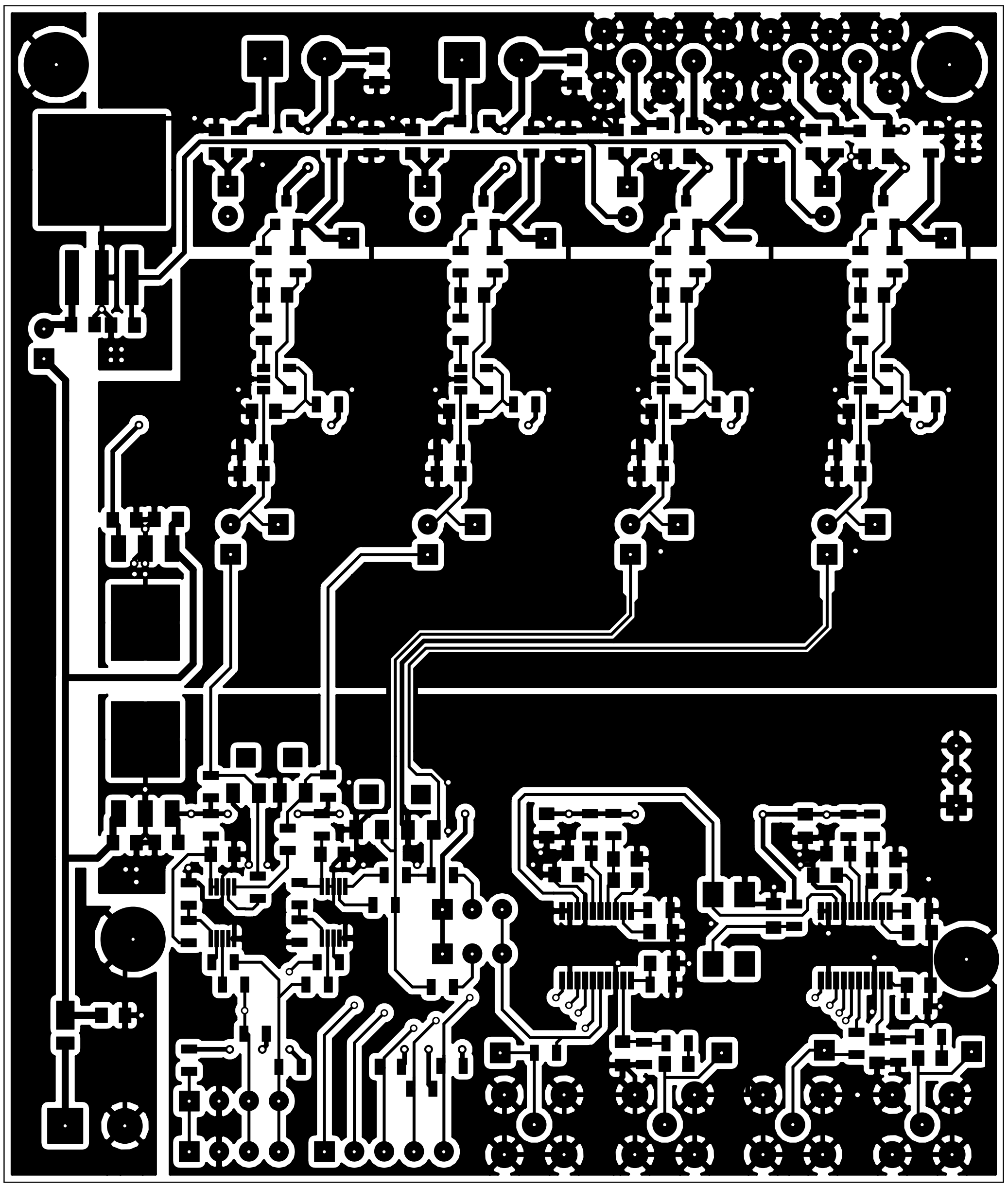
Sheet: /Interface/
File: Interface.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 4/7
MCP46X2
U9
I2C address: 0x2D
U8
I2C address: 0x2E
AD9834
U2
U3
V_freq
V_freq_0
V_freq_1
V_coil_0
V_coil_1
POT_TRIM_SMD
RV1
RV2
RV3
RV4
Jumper_NC_Dual
JP6
JP7
SCL
SDA
Vdd_IC
GNDD
GNDA
sin_inphase
sin_quadratur
pulse_inphase
pulse_quadratur
f_inphase
f_quadratur
J19
J20
J28
J29
SCLK
SDATA
CS_DDS_I
CS_DDS_Q
DSS_reset
Y1
Crystal_GND2_EN1

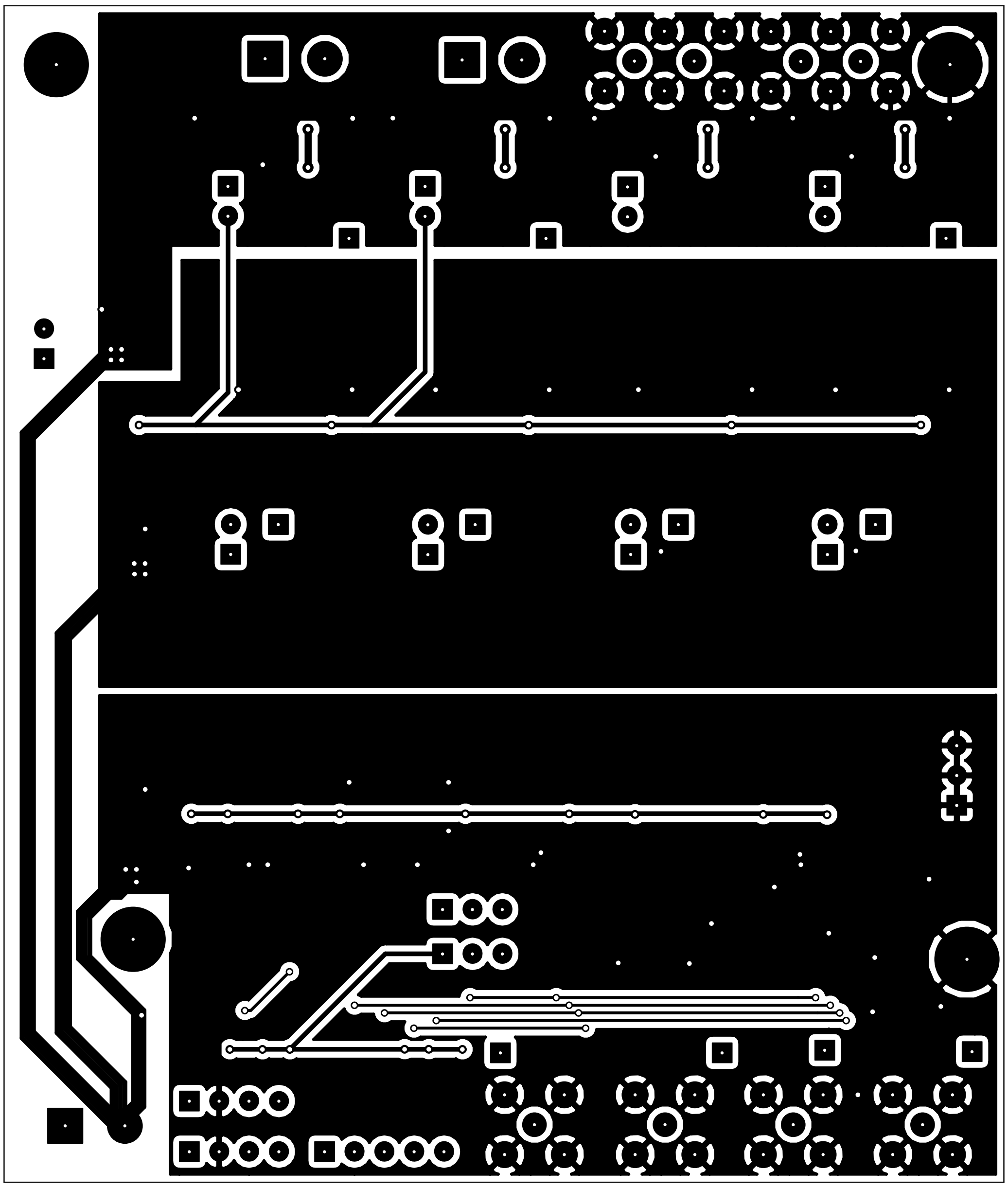

+9V_Driver
JP5
Jumper
C18
10uF
GNDPWR
R30
20R
LD_anode
LD_cathode
D2
Q2
NTR4170N
C46
1nF
J14
TEST_1P
R31
5R
GNDPWR
R28
1k
C17
1nF
R29
150R
R33
1k
U7
LMV793MF
GNDA
C16
1uF
GNDA
R27
10R
+5V_Driver
R52
ground_conn
R26
1M
GNDA
C11
1pF
GNDA
J13
TEST_1P
JP4
Jumper
V_set
Sheet: /Drive2/
File: Driver2.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 5/7

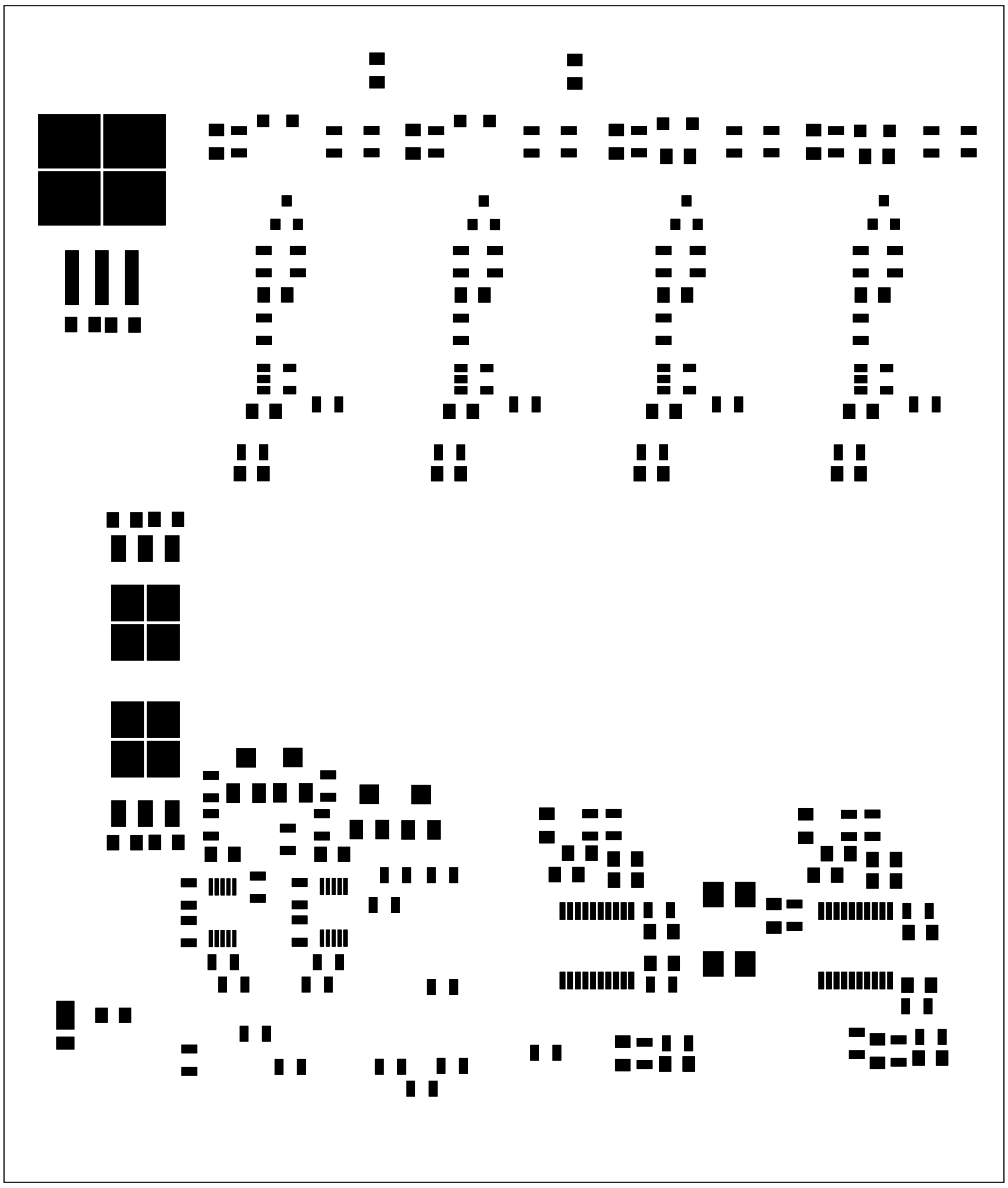

+5V_Driver
JP9
Jumper
C40
10uF
GNDPWR
R59
20R
coil+
coil-
D4
D
Q3
NTR4170N
C47
1nF
J22
TEST_1P
R60
5R
GNDPWR
R57
1k
C39
1nF
R58
150R
R56
1k
U10
LMV793MF
GNDA
C38
1uF
GNDA
R54
10R
+5V_Driver
R55
ground_conn
R53
1M
GNDA
C37
1pF
GNDA
J21
TEST_1P
JP8
Jumper
V_set
Sheet: /DriveCoil1/
File: DriverCoil1.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 6/7



+5V_Driver
JP11
Jumper
C44
10uF
GNDPWR
R67
20R
D5
coil+
coil−
Q4
NTR4170N
C48
1nF
J24
TEST_1P
R68
5R
GNDPWR
R65
1k
R66
150R
C43
1nF
R64
1k
U11
LMV793MF
GNDA
C42
1uF
GNDA
R62
10R
R63
ground_conn
R61
1M
GNDA
C41
1pF
GNDA
J23
TEST_1P
JP10
Jumper
+5V_Driver
V_set
Sheet: /DriveCoil2/
File: DriverCoil2.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 7/7







B Lock-In Amplifier Schematic

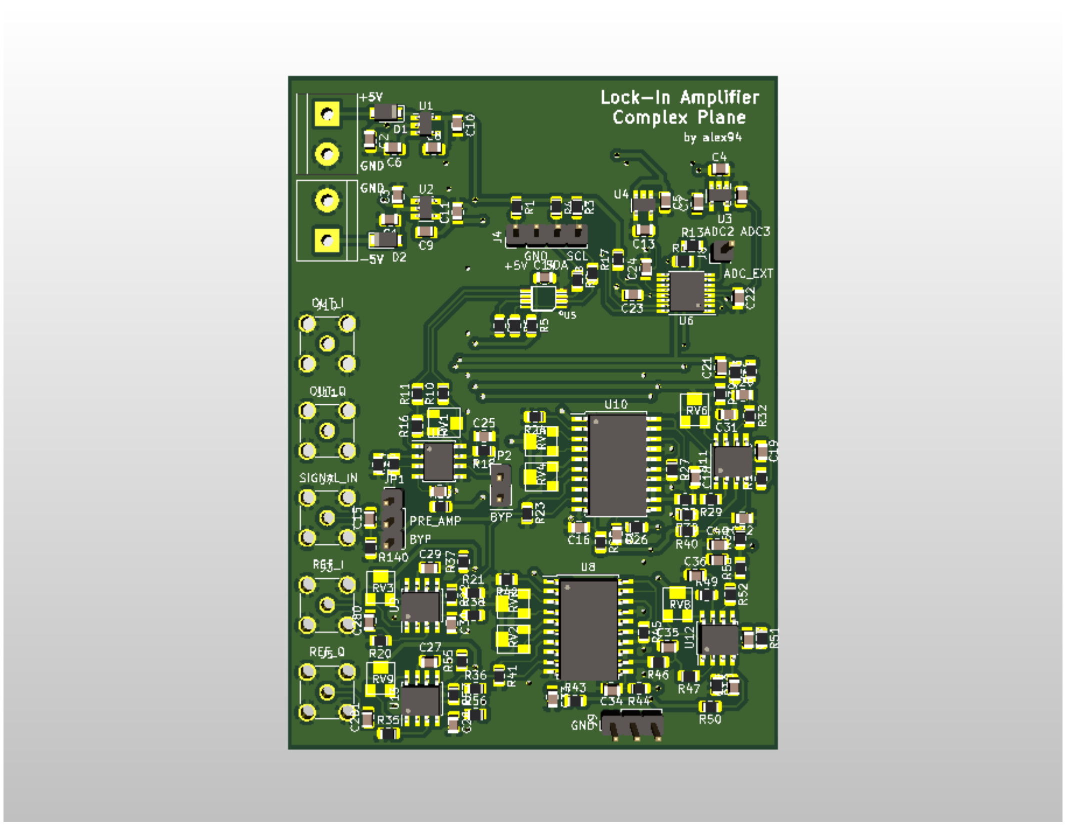
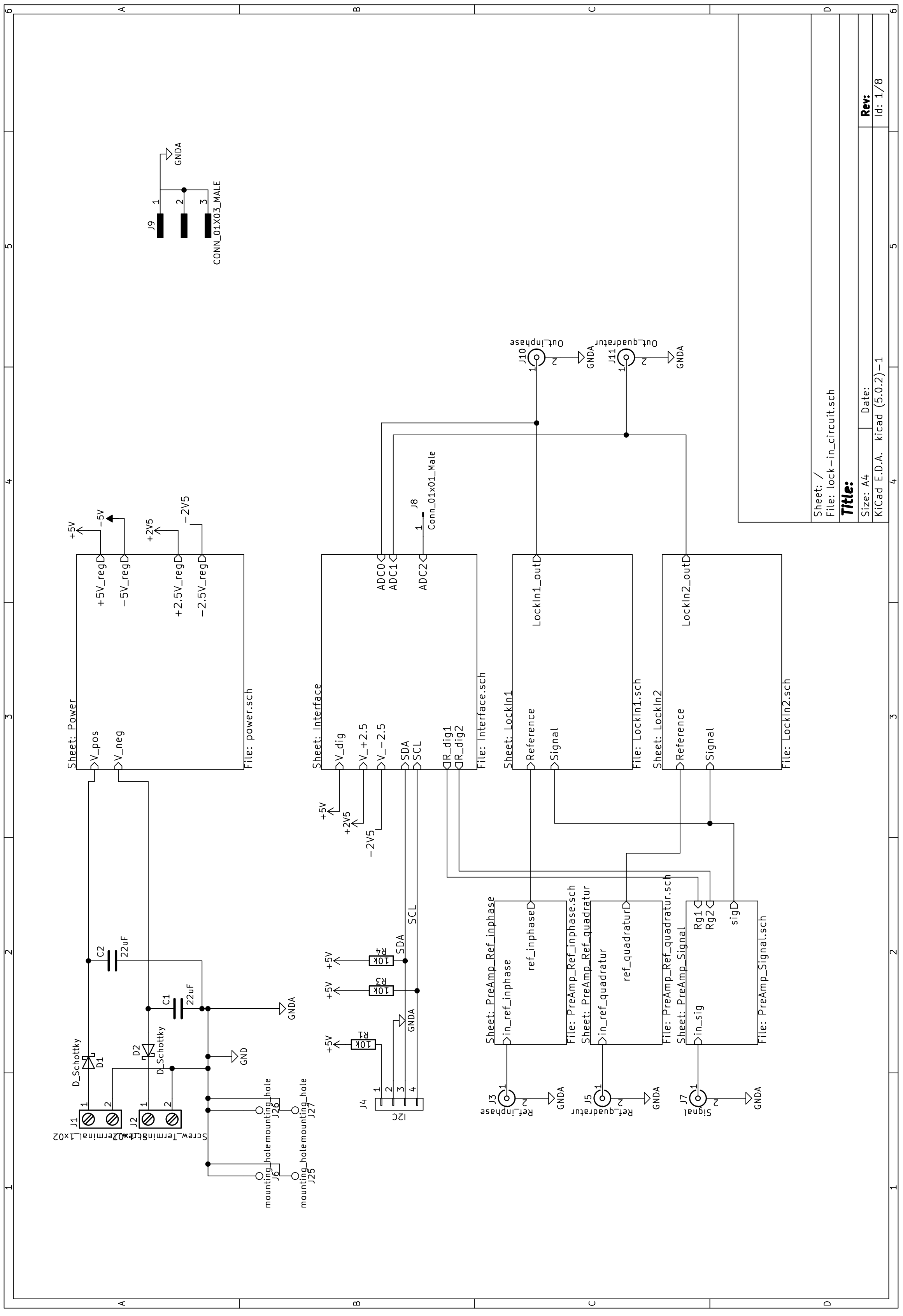
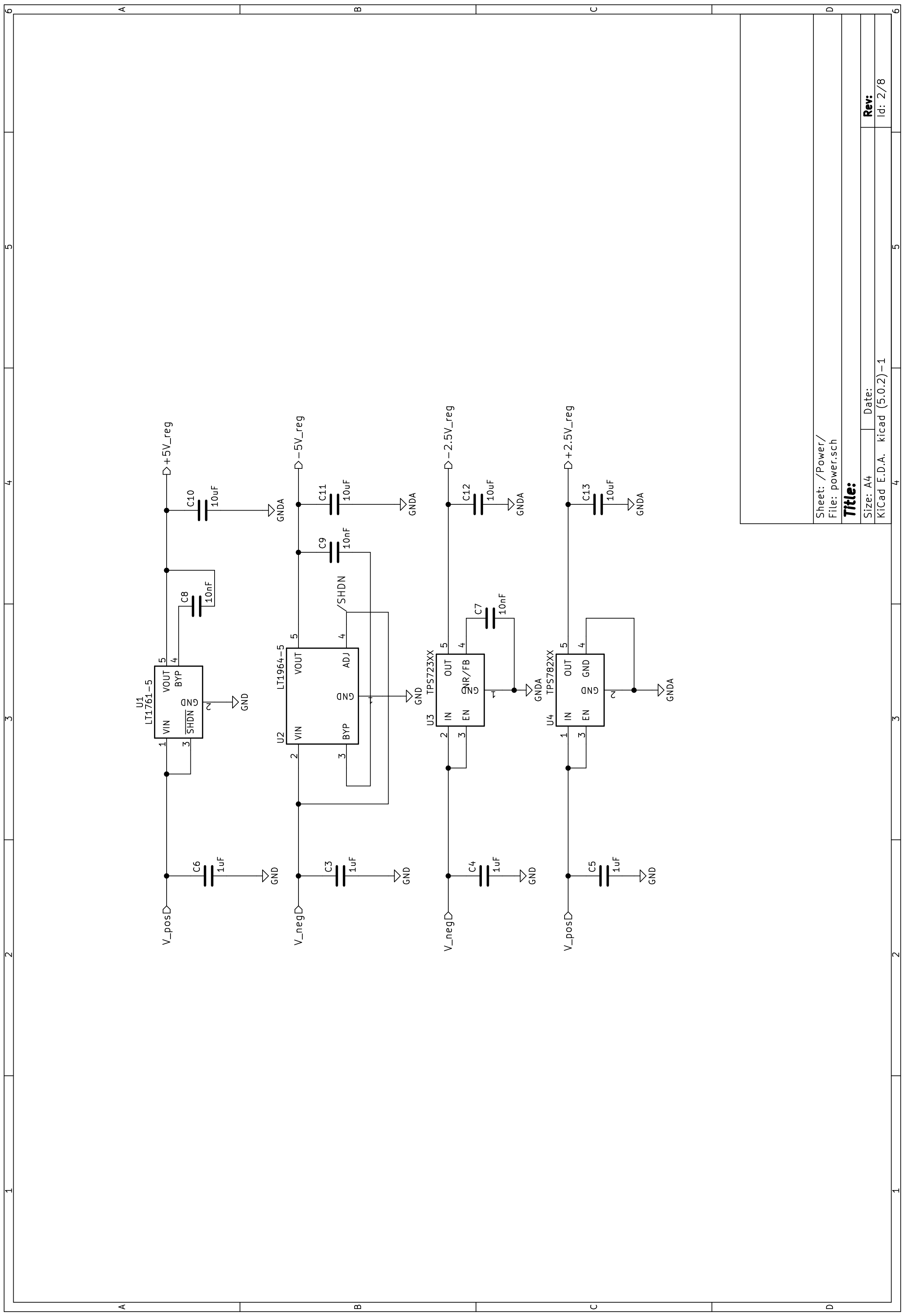
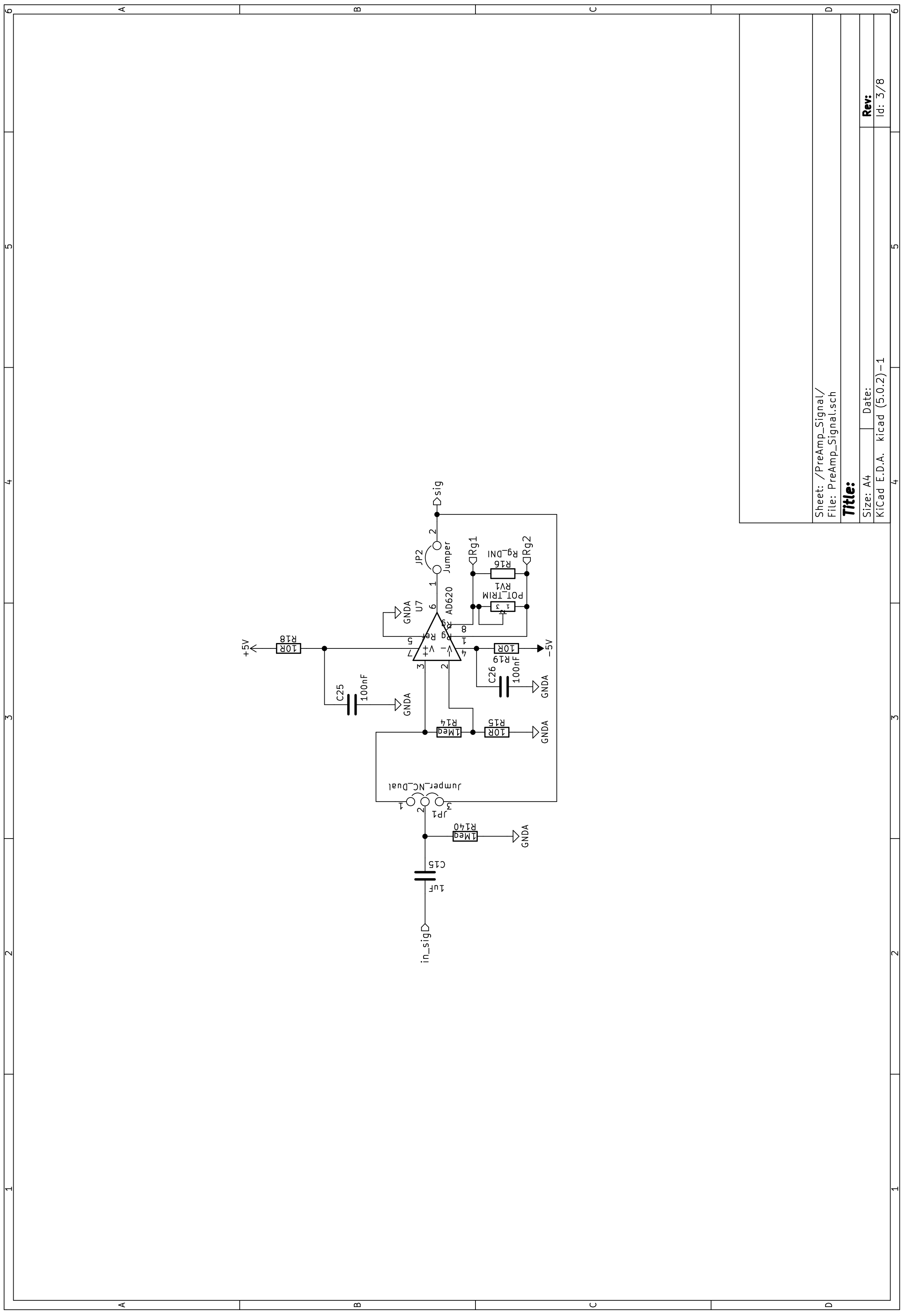


Figure B1: Lock-in amplifier schematic in 3D-view

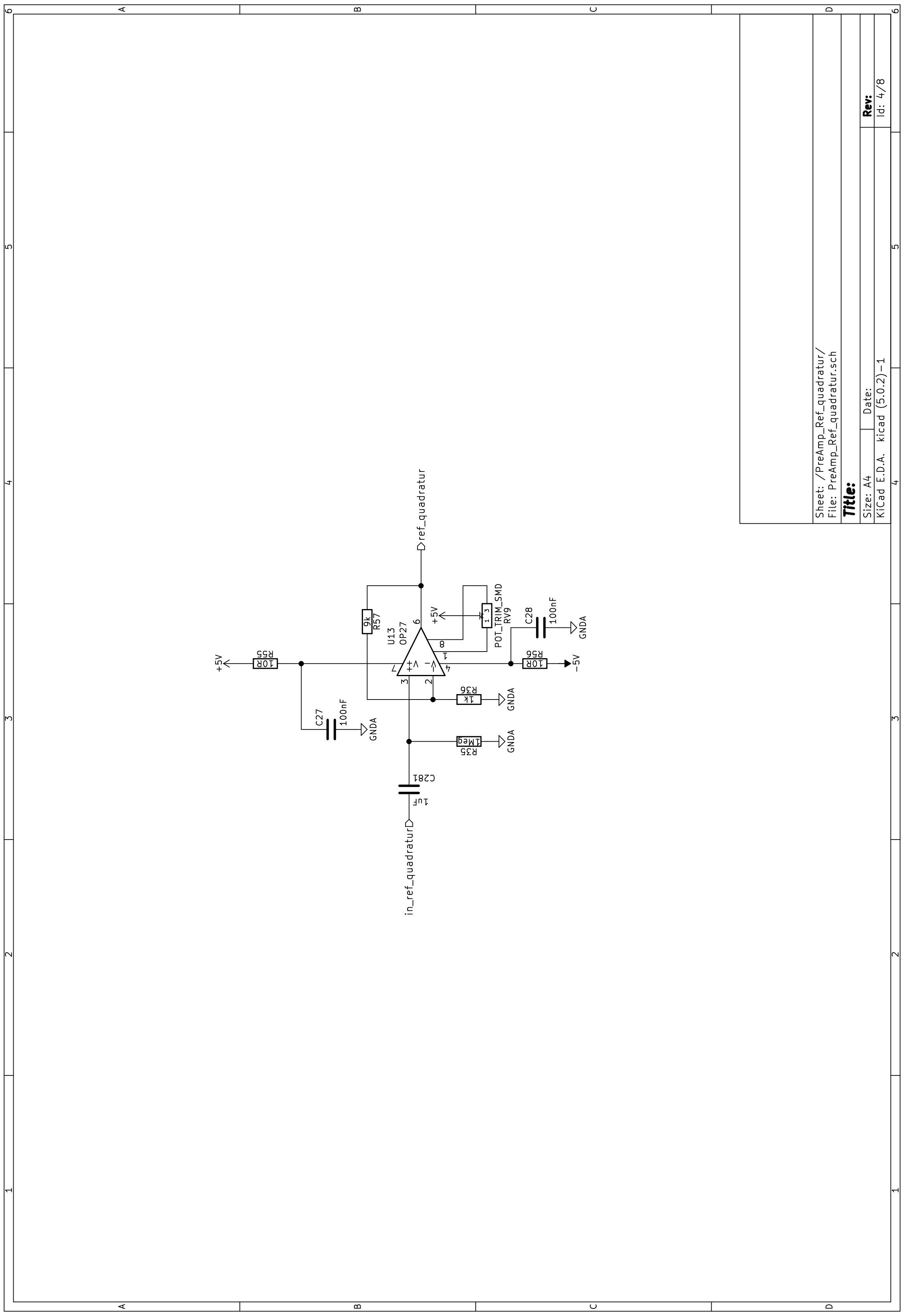


Sheet: /
File: lock-in_circuit.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 1/8
Sheet: Power
File: power.sch
V_pos
V_neg
+5V_reg
-5V_reg
+2.5V_reg
-2.5V_reg
Sheet: Interface
File: Interface.sch
V_dig
V_+2.5
V_-2.5
SDA
SCL
R_dig1
R_dig2
ADC0
ADC1
ADC2
Sheet: LockIn1
File: LockIn1.sch
Reference
Signal
LockIn1_out
Sheet: LockIn2
File: LockIn2.sch
LockIn2_out
Sheet: PreAmp_Ref_inphase
File: PreAmp_Ref_inphase.sch
in_ref_inphase
ref_inphase
Sheet: PreAmp_Ref_quadratur
File: PreAmp_Ref_quadratur.sch
in_ref_quadratur
ref_quadratur
Sheet: PreAmp_Signal
File: PreAmp_Signal.sch
in_sig
Rg1
Rg2
sig
J9
CONN_01X03_MALE
GNDA
J8
Conn_01x01_Male
J10
Out_inphase
J11
Out_quadratur
J3
Ref_inphase
J5
Ref_quadratur
J7
Signal
J4
I2C
R1
R3
R4
10K
+5V
-5V
+2V5
-2V5
GND
C1
C2
22uF
D1
D2
D_Schottky
J1
J2
Screw_Terminal_1x02
J6
J25
J26
J27
mounting_hole

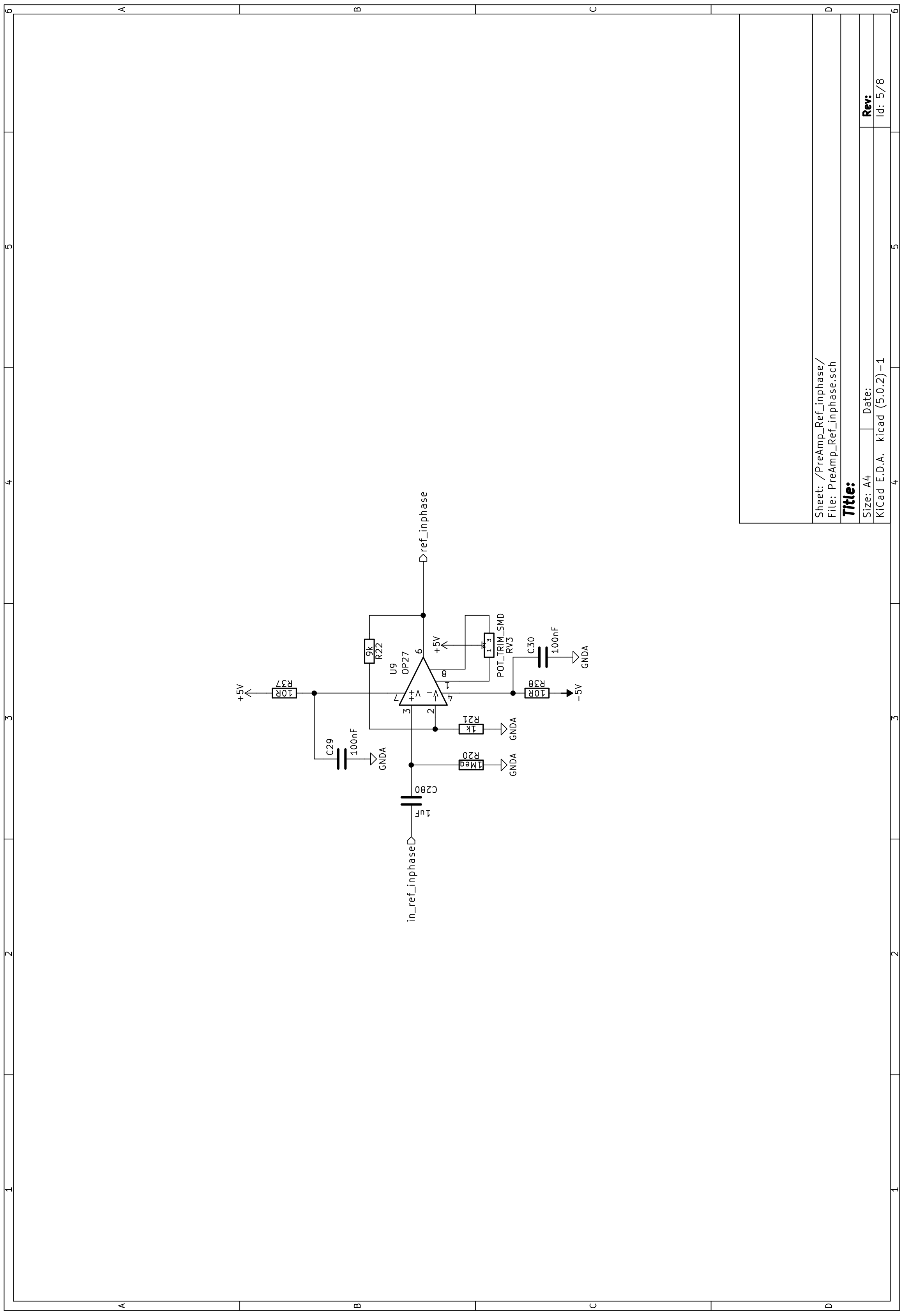

+5V_reg
-5V_reg
-2.5V_reg
+2.5V_reg
V_pos
V_neg
V_neg
V_pos
U1
LT1761-5
VIN
VOUT
BYP
GND
SHDN
U2
LT1964-5
VIN
VOUT
ADJ
GND
BYP
SHDN
U3
TPS723XX
IN
OUT
NR/FB
GND
EN
U4
TPS782XX
IN
OUT
GND
EN
C10
10uF
C11
10uF
C12
10uF
C13
10uF
C8
10nF
C9
10nF
C7
10nF
C6
1uF
C3
1uF
C4
1uF
C5
1uF
GNDA
GND
Sheet: /Power/
File: power.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 2/8

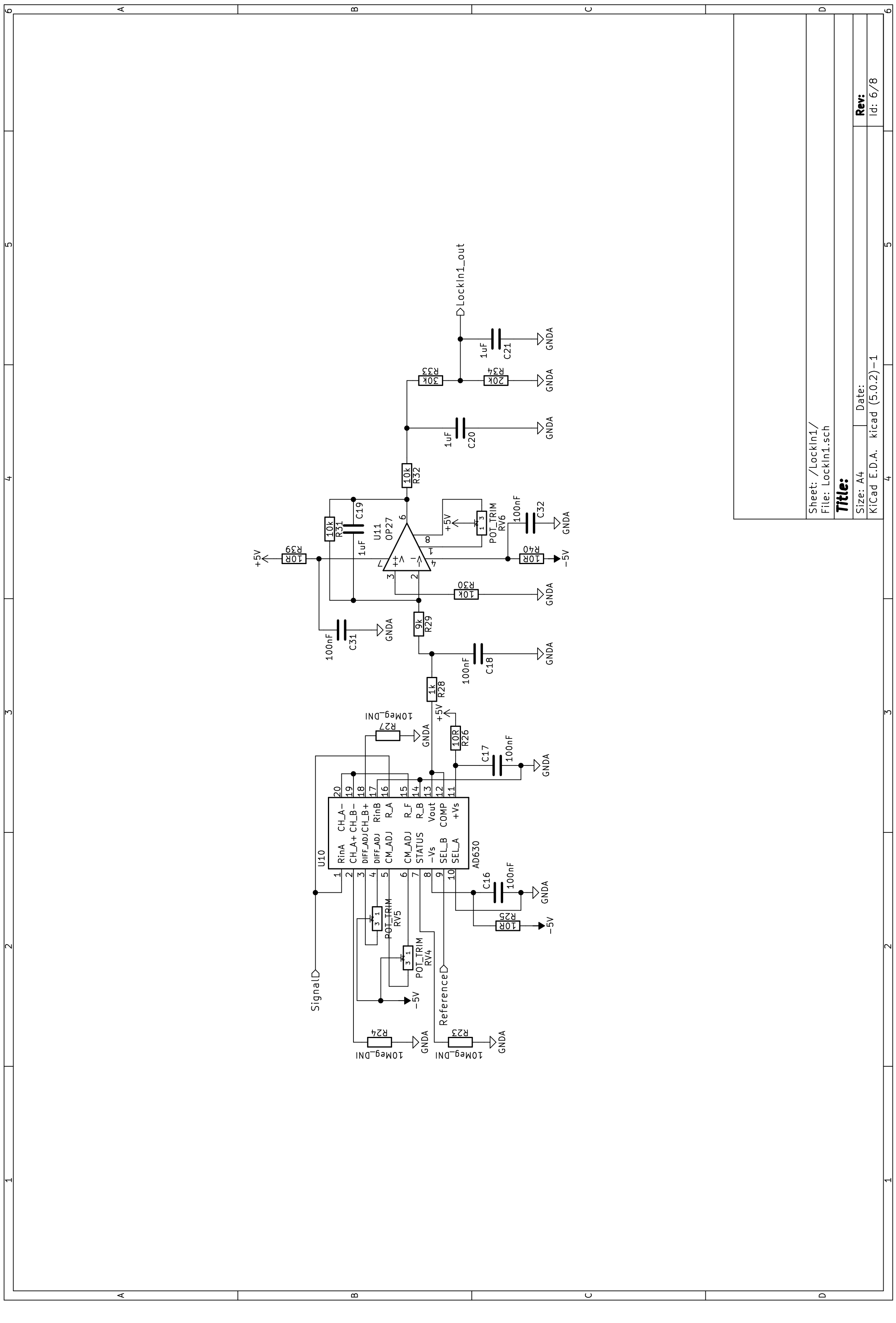


sig
JP2
Jumper
U7
AD620
GNDA
Rg1
Rg2
R16
Rg_DNI
RV1
POT_TRIM
+5V
R18
10R
-5V
R19
10R
C26
100nF
C25
100nF
R14
1Meg
R15
10R
Jumper_NC_Dual
JP1
R140
1Meg
C15
1uF
in_sig
Sheet: /PreAmp_Signal/
File: PreAmp_Signal.sch
Title:
Size: A4
Date:
KiCad E.D.A. kicad (5.0.2)-1
Rev:
Id: 3/8

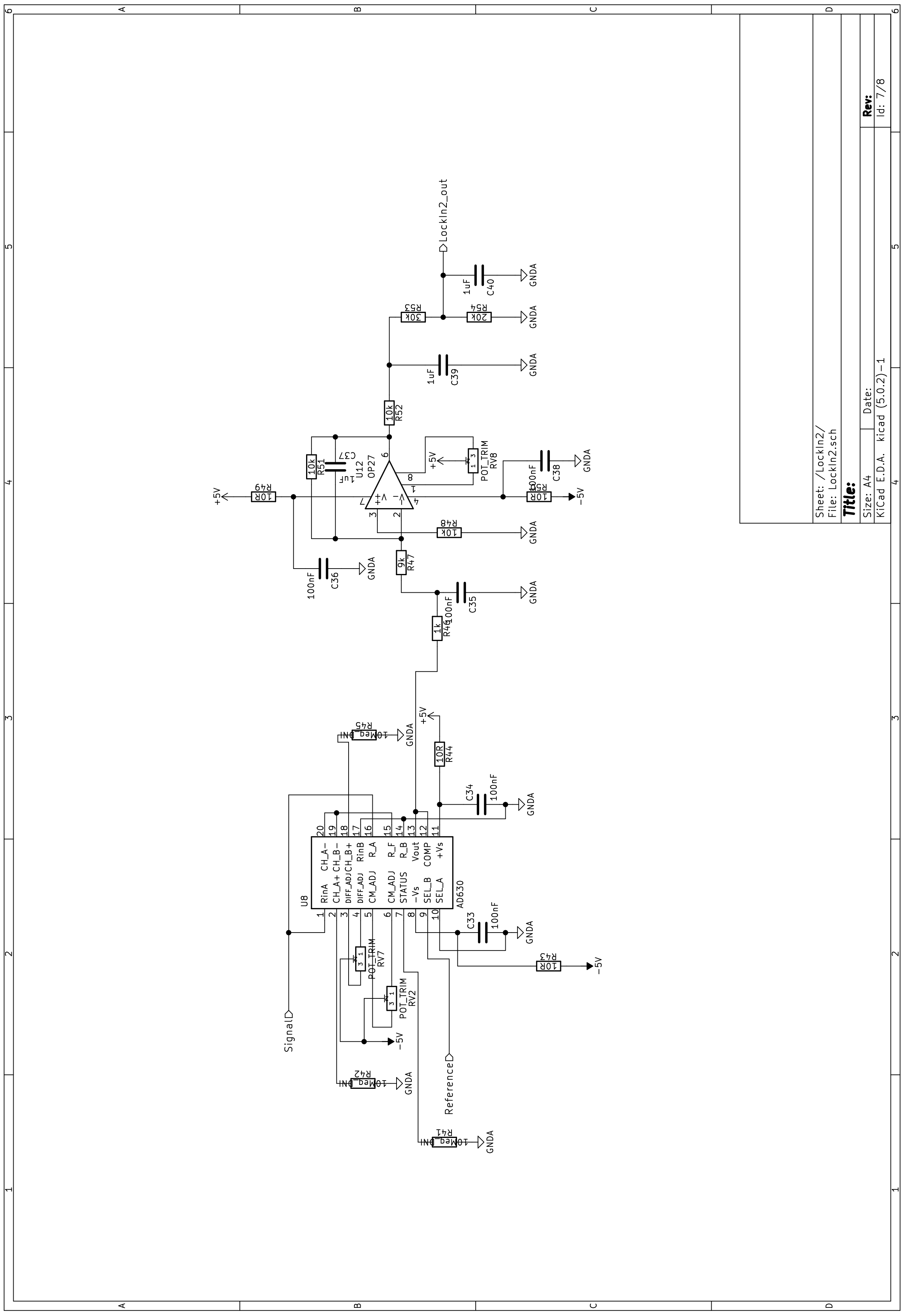


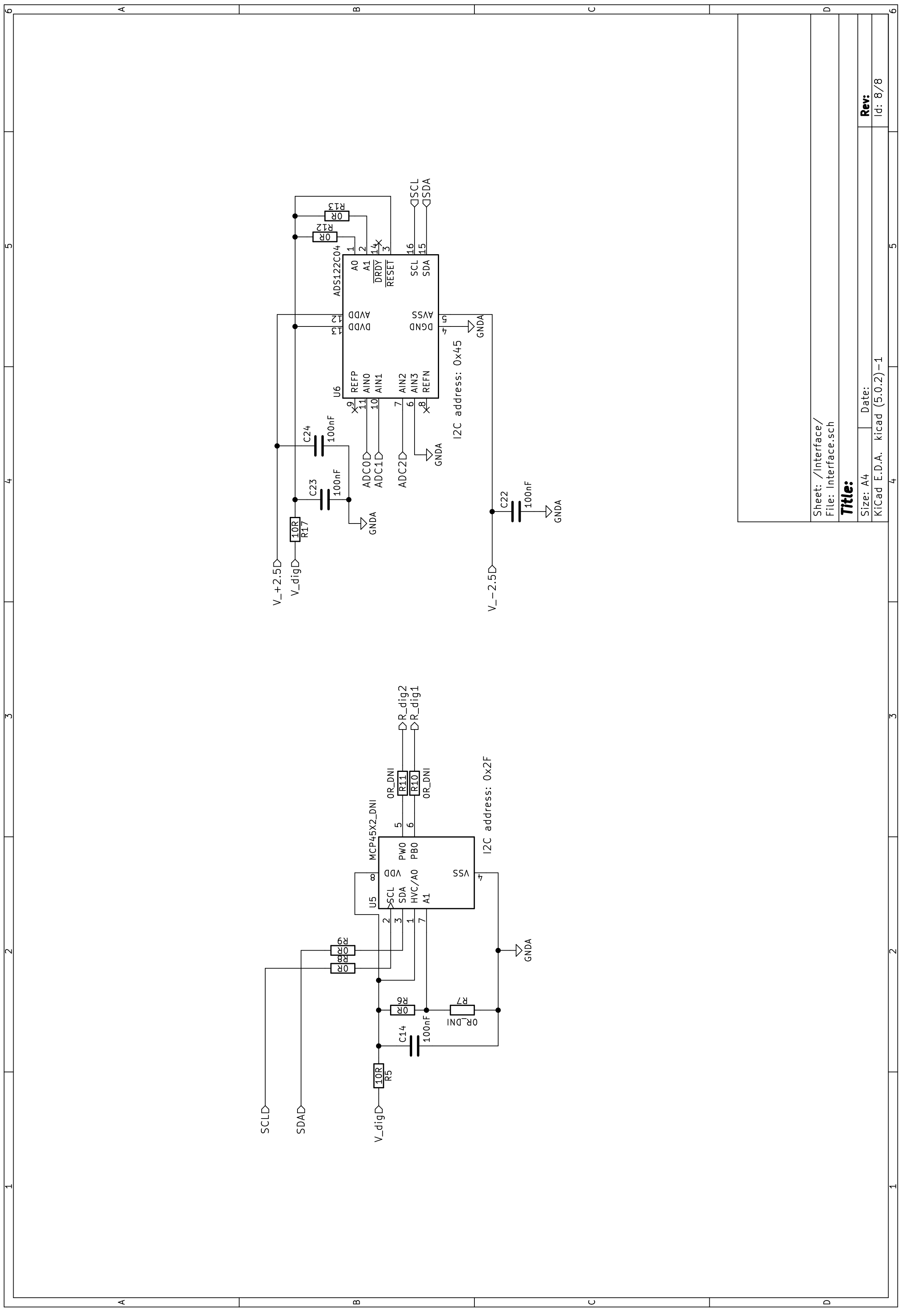
ref_quadratur
in_ref_quadratur
U13
OP27
R57
9k
POT_TRIM_SMD
RV9
+5V
-5V
R55
10R
R56
10R
C27
100nF
C28
100nF
R36
1k
R35
1Meg
C281
1uF
GNDA
Sheet: /PreAmp_Ref_quadratur/
File: PreAmp_Ref_quadratur.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 4/8


Sheet: /PreAmp_Ref_inphase/
File: PreAmp_Ref_inphase.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 5/8
ref_inphase
in_ref_inphase
U9
OP27
R22
9k
R37
10R
R38
10R
R21
1k
R20
1Meg
C29
100nF
C30
100nF
C280
1uF
RV3
POT_TRIM_SMD
+5V
-5V
GNDA

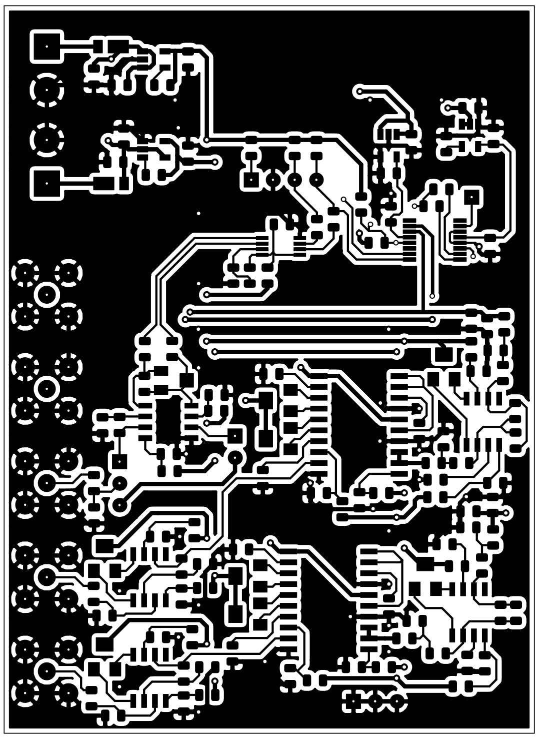


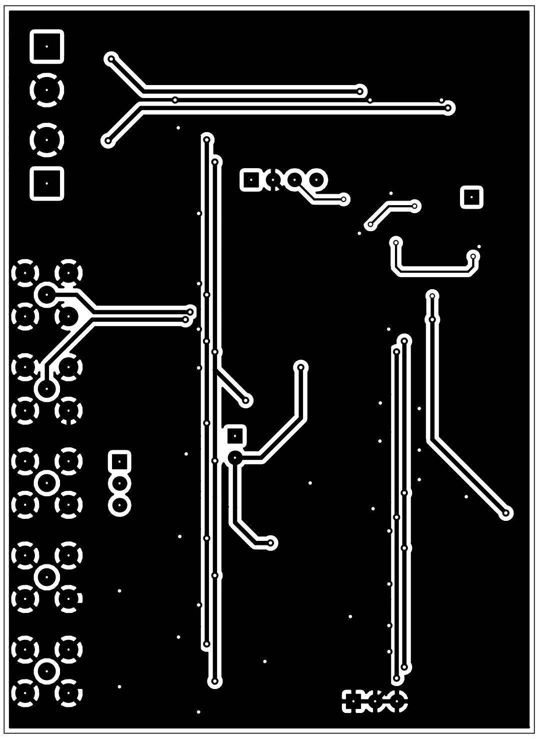
Sheet: /LockIn1/
File: LockIn1.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 6/8
LockIn1_out
Signal
Reference
U10
AD630
U11
OP27
RinA
CH_A+
DIFF_ADJ
DIFF_ADJ
CM_ADJ
CM_ADJ
STATUS
-Vs
SEL_B
SEL_A
CH_A-
CH_B-
CH_B+
RinB
R_A
R_F
R_B
Vout
COMP
+Vs
POT_TRIM
RV4
RV5
RV6
R23 10Meg_DNI
R24 10Meg_DNI
R27 10Meg_DNI
R25 10R
R26 10R
R39 10R
R40 10R
R28 1k
R29 9k
R30 10k
R31 10k
R32 10k
R33 30k
R34 20k
C16 100nF
C17 100nF
C18 100nF
C31 100nF
C32 100nF
C19 1uF
C20 1uF
C21 1uF
GNDA
+5V
-5V

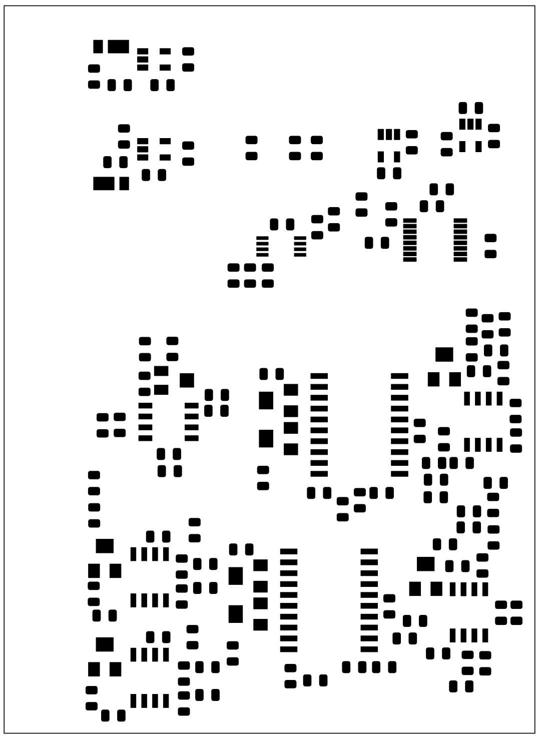

Sheet: /LockIn2/
File: LockIn2.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 7/8
LockIn2_out
Signal
Reference
U8
AD630
U12
OP27
RV7
RV2
RV8
POT_TRIM
R41
R42
R43
R44
R45
R46
R47
R48
R49
R50
R51
R52
R53
R54
C33
C34
C35
C36
C37
C38
C39
C40
GNDA
+5V
-5V



SCL
SDA
R13
0R
R12
0R
ADS122C04
A0
A1
DRDY
RESET
SCL
SDA
AVDD
DVDD
AVSS
DGND
GNDA
U6
REFP
AIN0
AIN1
AIN2
AIN3
REFN
I2C address: 0x45
C24
100nF
C23
100nF
R17
10R
ADC0
ADC1
ADC2
GNDA
V_+2.5
V_dig
C22
100nF
GNDA
V_-2.5
R_dig2
R_dig1
R11
0R_DNI
R10
0R_DNI
MCP45X2_DNI
PW0
PB0
VDD
VSS
SCL
SDA
HVC/A0
A1
U5
I2C address: 0x2F
R9
0R
R8
0R
GNDA
R6
0R
R7
0R_DNI
C14
100nF
R5
10R
SCL
SDA
V_dig
Sheet: /Interface/
File: Interface.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.2)-1
Id: 8/8







C Hardware Library

```
//// I2C Pins ////
// A4 (SDA)
// A5 (SCL)
//// SPI Pins DDS ////
// 6 MOSI
// 3 MISO
// 5 SCK
// 7 RESET
// 8 CS_I
// 9 CS_Q

#include <Wire.h>
#include <SPI.h>
#include <bitBangedSPI.h>


#define ACK "ACK"
#define NACK "NACK"
#define MCLK 8000000     // 8MHz clock frequency of DDS
#define I2C_TMP100 0x49
#define I2C_HDC1080 0x40
#define I2C_MCP4652_COIL 0x2E
#define I2C_MCP4652_LASER 0x2D
#define I2C_MCP4542 0x2F
#define MAX_RESIST 10000
#define I2C_ADS122C04 0x45
byte active_i2c = 0x00;

// Digital Pin Connects
#define W5100_CS  10
#define SDCARD_CS 4
#define DDS_RESET 7
#define CS_I 8
#define CS_Q 9
#define DDS_MOSI 6
#define DDS_MISO 3
#define DDS_SCK 5
#define DDS_SPI_MODE SPI_MODE2

bitBangedSPI bbSPI = bitBangedSPI(DDS_MOSI, DDS_MISO, DDS_SCK);

// Global Variables
bool i2c_status = false;

float dds_freq = 32768;       // resonance freq @ 32,768kHz
float ddsi_phase = 0.0;       // inphase DDS at 0? angle
float ddsq_phase = 90.0;      // quadrature DDS at 90? angle
byte ddsi_freg[4];
byte ddsq_freg[4];
byte ddsi_preg[2];
byte ddsq_preg[2];

int gain_ADS = 1;
// const float vref_ADS = 2.048;
const float vref_ADS = 2.5;
const long N_ADS = 8388608;       // 23-bit unsigned value of ADS
const long N_FREQ = 268435456;      // 28-bit value of frequency register (DDS)
const long N_PHASE = 4096;     // 12-bit value of phase register (DDS)
const long N_HDC1080 = 65536;   // 16-bit value of HDC1080
```

```

// Function Declarations
void initDDS();
void enableDDS();
void disableDDS();
void DDS_SetFreqInQuadrature(float freq);
void DDSI_SetFreqPhase(float freq, float phase);
void DDSQ_SetFreqPhase(float freq, float phase);
bool printHDC1080(int interval);
void initHDC1080();
void trigerHDC1080Measurement();
void readHDC1080(float *temperature, float *humidity);
float recalcHDC1080Temp(byte *buf);
float recalcHDC1080Hum(byte *buf);
bool printTMP100(int interval);
void initTMP100();
float readTMP100();
float recalcTMP100(byte *buf);
bool initADC();
bool setChannelGain(int channel, int gain);
bool startADConversion();
float readVoltage();
bool setCoil1_I(float mA);
bool setCoil2_I(float mA);
bool setLaser1_I(float mA);
bool setLaser2_I(float mA);
bool setAD620Gain(int gain);
bool setMCP4652_Resistance(byte address, int channel, unsigned long resist);
bool setMCP4542_Resistance(byte address, unsigned long resist);
bool setMCP4542_Binary(byte address, byte resist);
void spiWrite(int CS_pin, int len, byte *sequence);
bool i2cRead(byte address, byte reg, int len, byte *value);
bool i2cJustRead(byte address,int len, byte *value);
bool i2cWrite(byte address, byte reg, int len, byte *value);
void printI2Cstatus();
// End Function Declarations

void setupHardware()
{
  Wire.begin();          // join i2c bus (address optional for master)
  initTMP100();
  initHDC1080();
  setCoil1_I(0.0);
  setCoil2_I(0.0);
  setLaser1_I(0.0);
  setLaser2_I(0.0);

  initADC();

  // SPI Setup
  pinMode(SDCARD_CS, OUTPUT);
  digitalWrite(SDCARD_CS, HIGH);    // Deselect the SD card
  pinMode(DDS_RESET, OUTPUT);
  pinMode(CS_I, OUTPUT);
  pinMode(CS_Q, OUTPUT);
  digitalWrite(CS_I, HIGH);         // deactivate communication
  digitalWrite(CS_Q, HIGH);         // deactivate communication
  digitalWrite(DDS_RESET, HIGH);    // put into RESET Mode
  bbSPI.begin();

  initDDS();
}
```

```

/////////////////////////// AD9834 Implement ///////////////////////////////////

void initDDS()
{
  digitalWrite(DDS_RESET, HIGH);
  byte spi_buf[2];
  // control bits: consecutive writes, pin-controlled RESET
  spi_buf[0] = 0x22;
  // control bits: enable SIGN BIT OUT with DACs MSB bit, output = SIN
  spi_buf[1] = 0x28;
  spiWrite(CS_I, 2, spi_buf);    // puts DDS into RESET!!
  spiWrite(CS_Q, 2, spi_buf);    // puts DDS into RESET!!
}

void enableDDS()
{
  digitalWrite(DDS_RESET, LOW);
}

void disableDDS()
{
  digitalWrite(DDS_RESET, HIGH);
}

// freq between 30mHz and 8MHz - 30mHz steps
void DDS_SetFreqInQuadrature(float freq)
{
  digitalWrite(DDS_RESET, HIGH);
  DDSI_SetFreqPhase(freq, 0.0);
  DDSQ_SetFreqPhase(freq, 90.0);
  digitalWrite(DDS_RESET, LOW);
}

// freq between 30mHz and 8MHz - 30mHz steps
// phase between 0deg and 360deg - 0.0015deg steps
void DDSI_SetFreqPhase(float freq, float phase)
{
  // calculating FREQ_REG and PHASE_REG
  unsigned long freq_reg = N_FREQ * freq / MCLK;
  unsigned long phase_reg = N_PHASE * phase / 360;

  ddsi_freg[0] = byte(freq_reg & 0xFF);          // 1st 8 LSBs
  freq_reg >>= 8;                                // shift freq_reg by 8
  ddsi_freg[1] = byte(freq_reg & 0xFF);          // 2st 8 LSBs
  freq_reg >>= 8;                                // shift freq_reg by 8
  ddsi_freg[2] = byte(freq_reg & 0xFF);          // 1st 8 MSBs
  freq_reg >>= 8;                                // shift freq_reg by 8
  ddsi_freg[3] = byte(freq_reg & 0x0F);          // 2st 4 MSBs
  freq_reg >>= 8;                                // shift freq_reg by 8
  ddsi_preg[0] = byte(phase_reg & 0xFF);         // 8 MSBs
  phase_reg >>= 8;                               // shift phase_reg by 8
  ddsi_preg[1] = byte(phase_reg & 0x0F);         // 4 MSBs
  phase_reg >>= 8;                               // shift phase_reg by 8

  // commands and SPI Write
  int spi_line_len = 2;                // command line length is 2 bytes
  int spi_line_num = 5;                // 5 command lines to be send
  int len = spi_line_len*spi_line_num;
  byte spi_buf[10];
  // control bits: consecutive writes, pin-controlled RESET
  spi_buf[0] = 0x22;
  // control bits: enable SIGN BIT OUT with DACs MSB bit, output = SIN
  spi_buf[1] = 0x28;
  // 2nd LSB, FREQ0 register addressed (0x40)
  spi_buf[2] = (ddsi_freg[1] & 0x3F) | 0x40;
  spi_buf[3] = ddsi_freg[0];                                 // 1st LSB
  // 2nd MSB, FREQ0 register addressed (0x40)
  spi_buf[4] = (((ddsi_freg[3]<<2) | (ddsi_freg[2]>>6)) & 0x3F) | 0x40;
  spi_buf[5] = ((ddsi_freg[2]<<2) | (ddsi_freg[1]>>6));              // 1st MSB
  // 2nd MSB, PHASE0 register addressed (0xC0)
  spi_buf[6] = (ddsi_preg[1] & 0x0F) | 0xC0;
  spi_buf[7] = ddsi_preg[0];
  // control bits: consecutive writes, pin-controlled RESET  // 1st MSB
  spi_buf[8] = 0x22;
  // control bits: enable SIGN BIT OUT with DACs MSB bit, output = SIN
  spi_buf[9] = 0x28;
  spiWrite(CS_I, spi_line_len*spi_line_num, spi_buf);
}

// freq between 30mHz and (8MHz - 30mHz)
// phase between 0deg and (360deg - 0.0015deg)
void DDSQ_SetFreqPhase(float freq, float phase)
{
  // calculating FREQ_REG and PHASE_REG
  unsigned long freq_reg = N_FREQ * freq / MCLK;
  unsigned long phase_reg = N_PHASE * phase / 360;
  ddsq_freg[0] = byte(freq_reg & 0xFF);          // 1st 8 LSBs
  freq_reg >>= 8;                                // shift freq_reg by 8
  ddsq_freg[1] = byte(freq_reg & 0xFF);          // 2st 8 LSBs
  freq_reg >>= 8;                                // shift freq_reg by 8
  ddsq_freg[2] = byte(freq_reg & 0xFF);          // 1st 8 MSBs
  freq_reg >>= 8;                                // shift freq_reg by 8
  ddsq_freg[3] = byte(freq_reg & 0x0F);          // 2st 4 MSBs
  freq_reg >>= 8;                                // shift freq_reg by 8
  ddsq_preg[0] = byte(phase_reg & 0xFF);         // 8 MSBs
  phase_reg >>= 8;                               // shift phase_reg by 8
  ddsq_preg[1] = byte(phase_reg & 0x0F);         // 4 MSBs
  phase_reg >>= 8;                               // shift phase_reg by 8

  // commands and SPI Write
  int spi_line_len = 2;      // command line length is 2 bytes
  int spi_line_num = 5;      // 5 command lines to be send
  int len = spi_line_len*spi_line_num;
  byte spi_buf[10];
  byte spi_buf[10];
  // control bits: consecutive writes, pin-controlled RESET
  spi_buf[0] = 0x22;
  // control bits: enable SIGN BIT OUT with DACs MSB bit, output = SIN
  spi_buf[1] = 0x28;
  // 2nd LSB, FREQ0 register addressed (0x40)
  spi_buf[2] = (ddsq_freg[1] & 0x3F) | 0x40;
  spi_buf[3] = ddsq_freg[0];                                 // 1st LSB
  // 2nd MSB, FREQ0 register addressed (0x40)
  spi_buf[4] = (((ddsq_freg[3]<<2) | (ddsq_freg[2]>>6)) & 0x3F) | 0x40;
  spi_buf[5] = ((ddsq_freg[2]<<2) | (ddsq_freg[1]>>6));              // 1st MSB
  // 2nd MSB, PHASE0 register addressed (0xC0)
  spi_buf[6] = (ddsq_preg[1] & 0x0F) | 0xC0;
  spi_buf[7] = ddsq_preg[0];
  // control bits: consecutive writes, pin-controlled RESET  // 1st MSB
  spi_buf[8] = 0x22;
  // control bits: enable SIGN BIT OUT with DACs MSB bit, output = SIN
  spi_buf[9] = 0x28;
  spiWrite(CS_Q, spi_line_len*spi_line_num, spi_buf);
```

```
}

/////////////////////////////// HDC1080 Implement ////////////////////////////////

bool printHDC1080(int interval)
{
  float temperature = 0.0;
  float humidity = 0.0;
  readHDC1080(&temperature, &humidity);
  if(i2c_status)
  {
    Serial.print("Temp.: "); Serial.print(temperature, 4);
  Serial.print(" degC, Hum.: "); Serial.print(humidity, 4);
  Serial.print(" %RH (HDC1080)\n");
  }
  else
    printI2Cstatus();
  delay(interval);
  return i2c_status;
}

void initHDC1080()
{
  byte cmd[2];
  cmd[0] = 0x10;
  cmd[1] = 0x00;
  i2c_status = i2cWrite(I2C_HDC1080, 0x02, 2, cmd);
}

void trigerHDC1080Measurement()         // wait at least 7ms after triggering
{
  i2c_status = i2cWrite(I2C_HDC1080, 0x00, 0, NULL);
}

void readHDC1080(float *temperature, float *humidity)
{
  byte buf[4];
  int buf_len = 4;
  // read 2 byte temperature value followed by 2 byte humidity
  i2c_status = i2cJustRead(I2C_HDC1080, buf_len, buf);
  *temperature = recalcHDC1080Temp(buf);
  *humidity = recalcHDC1080Hum(buf);
}

float recalcHDC1080Temp(byte *buf)
{
  float temp = (256 * buf[0]) + buf[1];
  temp = temp/N_HDC1080;
  temp = temp*165 - 40;        // temperature in degC
  return temp;
}

float recalcHDC1080Hum(byte *buf)
{
  float hum = unsigned((256 * buf[2]) + buf[3]);
  hum = hum/N_HDC1080;
  hum = hum*100;               // humidity in %RH
  return hum;
}

//////////////////////////////// TMP100 Implement ///////////////////////////////

```

```
bool printTMP100(int interval)
{
  float temperature = 0.0;
  temperature = readTMP100();
  if(i2c_status)
  {
    Serial.print("Temp.: "); Serial.print(temperature, 4);
  Serial.print(" degC (TMP100)\n");
  }
  else
    printI2Cstatus();
  delay(interval);
  return i2c_status;
}

void initTMP100()
{
  byte cmd = 0x60;
  i2c_status = i2cWrite(I2C_TMP100, 0x01, 1, &cmd);      // config TMP100
}

float readTMP100()
{
  byte buf[2];
  int buf_len = 2;
  // read 2 byte temperature value
  i2c_status = i2cRead(I2C_TMP100, 0x00, buf_len, buf);
  return recalcTMP100(buf);
}

float recalcTMP100(byte *buf)
{
  buf[1] >>= 4;
  float temp = 16.0 * buf[0] + buf[1];
  temp *= 0.0625;
  if (temp > 127.9375)
    temp -= 2 * 128.0;
  return temp;
}

///////////////////////////// ADS122C04 Implement //////////////////////////////

bool initADC()
{
  byte buf[3];
  i2c_status = i2cWrite(I2C_ADS122C04, 0x06, 0, NULL);     // RESET command
  delay(1);
  buf[0] = 0x21;  // reg 00h value
  buf[1] = 0x44;  // reg 01h address
  buf[2] = 0xDC;  // reg 01h value
  // set ADC0 and gain 1, PGA disabled, continuous mode, 2000SPS (turbo mode)
  i2c_status &= i2cWrite(I2C_ADS122C04, 0x40, 3, buf);
  return i2c_status;
}

bool setChannelGain(int channel, int gain)
{
  if( ((channel > 2) || (channel < 0))
    || ((gain != 1) && (gain != 2) && (gain != 4)) )
    return false;
  gain_ADS = gain;
  byte buf[1];
  // set channel
  if(channel == 0)
    buf[0] = 0x20;
  if(channel == 1)
    buf[0] = 0x50;
  if(channel == 2)
    buf[0] = 0x60;
  // set gain
  if(gain == 1)
    buf[0] += 0;
  if(gain == 2)
    buf[0] += 2;
  if(gain == 4)
    buf[0] += 4;
  // bypass PGA
  buf[0] += 1;
  i2c_status = i2cWrite(I2C_ADS122C04, 0x40, 1, buf); // 0x40 = reg 00h address
  i2c_status &= startADConversion();                  // start conversion
  return i2c_status;
}

bool startADConversion()
{
  i2c_status = i2cWrite(I2C_ADS122C04, 0x08, 0, NULL);
  return i2c_status;
}

float readVoltage()
{
  byte buf[3];
  float temp_volt = 0;
  i2c_status = i2cRead(I2C_ADS122C04, 0x10, 3, buf);
  if(!i2c_status)
    return temp_volt;

  unsigned long temp_bin = (buf[0] & 0x7F);
  temp_bin <<= 8;
  temp_bin += buf[1];
  temp_bin <<= 8;
  temp_bin += buf[2];
  temp_volt = 2.0*vref_ADS / (float(N_ADS)*float(gain_ADS))*(float(temp_bin));
  if(buf[0] & 0x80)
    temp_volt = temp_volt - 2.0*vref_ADS;
  return temp_volt;
}

////////////////////////////// PCB current control /////////////////////////////

bool setCoil1_I(float mA)
{
  long resist = mA * 100;      // 10k =~ 100mA, 100R =~ 1mA
  if((resist > MAX_RESIST) || (resist < 0))
    return false;

  i2c_status = setMCP4652_Resistance(I2C_MCP4652_COIL, 0, resist);
  return i2c_status;
}

bool setCoil2_I(float mA)
{
  long resist = mA * 100;      // 10k =~ 100mA, 100R =~ 1mA
  if((resist > MAX_RESIST) || (resist < 0))
```

```
    return false;

  i2c_status = setMCP4652_Resistance(I2C_MCP4652_COIL, 1, resist);
  return i2c_status;
}

bool setLaser1_I(float mA)
{
  long resist = mA * 100;      // 10k =~ 100mA, 100R =~ 1mA
  if((resist > MAX_RESIST) || (resist < 0))
    return false;

  i2c_status = setMCP4652_Resistance(I2C_MCP4652_LASER, 0, resist);
  return i2c_status;
}

bool setLaser2_I(float mA)
{
  long resist = mA * 100;      // 10k =~ 100mA, 100R =~ 1mA
  if((resist > MAX_RESIST) || (resist < 0))
    return false;

  i2c_status = setMCP4652_Resistance(I2C_MCP4652_LASER, 1, resist);
  return i2c_status;
}

////////////////////////////// Input Gain Control //////////////////////////////

int gain_len = 18;
int gain_available[] = {1,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,30,50,106};
byte gain_array[] = {0x80,0x7E,0x69,0x5A,0x4F,0x46,0x3F,0x39,0x53,0x31,0x2D,
           0x2A,0x25,0x23,0x21,0x16,0x0D,0x06};

bool setAD620Gain(int gain)
{
  for(int i = 0; i < gain_len; i++)
  {
    if(gain == gain_available[i])
    {
      i2c_status = setMCP4542_Binary(I2C_MCP4542, gain_array[i]);
      return i2c_status;
    }
  }
  return false;
}

////////////////////////////// MCP4652 Implement //////////////////////////////

bool setMCP4652_Resistance(byte address, int channel, unsigned long resist)
{
  byte buf[2];
  if(channel == 0)
    buf[0] = 0x00;
  else if(channel == 1)
    buf[0] = 0x10;
  else
    return false;      // return false if channel >= 1

  if(resist == MAX_RESIST)
  {
    buf[0] |= 0x01;
    buf[1] = 0x00;
  }
  else if(resist < MAX_RESIST)
    buf[1] = byte(resist*256/MAX_RESIST);
  else
    return false;     // return false if resistance is too large... > MAX_RESIST

  i2c_status =  i2cWrite(address, buf[0], 1, &buf[1]);
  return i2c_status;
}

////////////////////////////// MCP4542 Implement //////////////////////////////

bool setMCP4542_Resistance(byte address, unsigned long resist)
{
  byte buf[2];
  buf[0] = 0x00;

  if(resist <= MAX_RESIST)
    buf[1] = byte(resist*128/MAX_RESIST);
  else
    return false;     // return false if resistance is too large... > MAX_RESIST

  i2c_status = i2cWrite(address, buf[0], 1, &buf[1]);
  return i2c_status;
}

bool setMCP4542_Binary(byte address, byte resist)
{
  byte buf[2];
  buf[0] = 0x00;
  buf[1] = resist;

  i2c_status = i2cWrite(address, buf[0], 1, &buf[1]);
  return i2c_status;
}

//////////////////////////// interface level /////////////////////////////////

///////// SPI Communication, write only /////////
void spiWrite(int CS_pin, int len, byte *sequence)
{
  digitalWrite(CS_pin, LOW);
  for (int i = 0; i < len; i++)
  {
    bbSPI.transfer(sequence[i]);
  }
  digitalWrite(CS_pin, HIGH);
}

///////// I2C Communication, read/write /////////
bool i2cRead(byte address, byte reg, int len, byte *value)
{
  active_i2c = address;
  Wire.beginTransmission(address);
  Wire.write(reg);
  Wire.endTransmission();

  Wire.requestFrom(address, len);
  int i = 0;
  while(Wire.available())
  {
    value[i] = Wire.read();
    i++;
  }
  if (i == len)
    i2c_status = true;
  else
    i2c_status = false;

  return i2c_status;
}

bool i2cJustRead(byte address,int len, byte *value)
{
  active_i2c = address;
  Wire.requestFrom(address, len);
  int i = 0;
  while(Wire.available())
  {
    value[i] = Wire.read();
    i++;
  }
  if (i == len)
    i2c_status = true;
  else
    i2c_status = false;

  return i2c_status;
}

bool i2cWrite(byte address, byte reg, int len, byte *value)
{
  active_i2c = address;
  Wire.beginTransmission(address);
  Wire.write(reg);
  if(len > 0)
    Wire.write(value, len);
  if(Wire.endTransmission() == 0)
    i2c_status = true;
  else
    i2c_status = false;
  return i2c_status;
}

void printI2Cstatus()
{
  Serial.print("0x");
  Serial.print(active_i2c,HEX);
  Serial.print(": ");
  if (i2c_status)
  {
    Serial.print(ACK);
  }
  else
  {
    Serial.print(NACK);
  }
  Serial.print("\n");
}
```

D Arduino Code

```
//// I2C Pins ////
// A4 (SDA)
// A5 (SCL)
//// SPI Pins DDS ////
// 6 MOSI
// 3 MISO
// 5 SCK
// 7 RESET
// 8 CS_I
// 9 CS_Q

#include <SimpleTimer.h>
SimpleTimer timer;

#include <Wire.h>
#include <SPI.h>
#include <hw_driver_serialOnly.h>
#include <stdio.h>
#include <ctype.h>

char str1[] = "frequency [Hz]; time [ms]; V_inphase [V]; V_quadrature [V]";
char str2[] = "phase [deg]; time [ms]; V_inphase [V]; V_quadrature [V]";
char str3[] = "focus current [mA]; time [ms]; V_inphase [V]; V_quadrature [V]";
char str4[] = "position current [mA]; time [ms]; V_inphase [V];
         V_quadrature [V]";

// Valiables for Mearement routine
float sweepStep = 0.0;               // in Hz

float startFrequency = 0.0;          // in Hz
float stopFrequency = 0.0;           // in Hz
float oldFrequency = 0.0;            // in Hz
float newFrequency = 0.0;            // in Hz
float lockedFrequency = 0.0;         // in Hz

float startPhase = 0.0;          // in Hz
float stopPhase = 0.0;           // in Hz
float oldPhase = 0.0;            // in Hz
float newPhase = 0.0;            // in Hz

float startCurrent = 0.0;          // in mA
float stopCurrent = 0.0;           // in mA
float oldCurrent = 0.0;            // in mA
float newCurrent = 0.0;            // in mA

int timerID_meas = 0;
unsigned long startTime = 0;         // in ms
unsigned long meas_points = 0;
unsigned long meas_counter = 0;
unsigned long decadic_counter = 0;
int time_counter = 0;

#define MODE_NO                      0
//#define MODE_TEMPERATURE           1
#define MODE_SWEEP_FREQ              2
#define MODE_SWEEP_PHASE             3
#define MODE_SINGLE_FREQ             4
#define MODE_SWEEP_FOCUS             5
#define MODE_SWEEP_POSITION          6
```

```
#define MODE_SWEEP_FREQ_LOCKED    7
#define MODE_SWEEP_FREQ_LOCKED_DECADIC    8
int meas_mode = MODE_NO;
int laser_current = 0.0;            // in mA
float volt_inphase = 0.0;           // in V
float volt_quadrature = 0.0;        // in V
const char meas_done[] = "measurement done!";

// Timer for Measurement routine, every 1 ms, 10 ms sampling
void measurementEvent()
{
  switch(time_counter)
  {
    case -1:
    {
      time_counter = 1;
    }
    case 0:     // set frequency and wait 6x1ms
    {
    // end the measurement routine when stop condition is fullfilled
      if(((newFrequency > stopFrequency) && (meas_mode == MODE_SWEEP_FREQ))
      || ((newPhase > stopPhase) && (meas_mode == MODE_SWEEP_PHASE))
      || ((newCurrent > stopCurrent) && ((meas_mode == MODE_SWEEP_FOCUS)
    || (meas_mode == MODE_SWEEP_POSITION)))
      || ((meas_counter >= meas_points) && (meas_mode == MODE_SINGLE_FREQ))
      || ((newFrequency > stopFrequency) &&
     (meas_mode == MODE_SWEEP_FREQ_LOCKED))
      || ((newFrequency > stopFrequency) &&
     (meas_mode == MODE_SWEEP_FREQ_LOCKED_DECADIC)))
      {
    // turn of output stages
        setLaser1_I(0.0);
        setLaser2_I(0.0);
        setCoil1_I(0.0);
        setCoil2_I(0.0);
        disableDDS();
        timer.disable(timerID_meas);  // turn of timer routine
        Serial.print(meas_done); Serial.print('\n');
      }
      // frequency sweep mode: set new frequency for both DDS
      if(meas_mode == MODE_SWEEP_FREQ)
      {
        if(newFrequency != oldFrequency)
        {
          setLaser1_I(0.0);
          DDS_SetFreqInQuadrature(newFrequency);
          oldFrequency = newFrequency;
          setLaser1_I(laser_current);
        }
        meas_counter++;
        if(meas_counter == meas_points)
        {
           meas_counter = 0;
           newFrequency = oldFrequency + sweepStep;
        }
      }
    // locked frequency sweep mode: set new frequency on second DDS,
    // first DDS to locked frequency
      else if((meas_mode == MODE_SWEEP_FREQ_LOCKED)
       || (meas_mode == MODE_SWEEP_FREQ_LOCKED_DECADIC))
      {
        if(newFrequency != oldFrequency)
        {
          setLaser1_I(0.0);
          disableDDS();
          DDSI_SetFreqPhase(lockedFrequency, 0);
          DDSQ_SetFreqPhase(newFrequency, 0.0);
          enableDDS();
          oldFrequency = newFrequency;
          setLaser1_I(laser_current);
        }
        meas_counter++;
        if(meas_counter == meas_points)
        {
           if(meas_mode == MODE_SWEEP_FREQ_LOCKED)
           {
             meas_counter = 0;
             newFrequency = oldFrequency + sweepStep;
           }
           if(meas_mode == MODE_SWEEP_FREQ_LOCKED_DECADIC)
           {
             meas_counter = 0;
             decadic_counter++;
             newFrequency = startFrequency * pow(10, sweepStep*decadic_counter);
           }
        }
      }
    // phase sweep mode: set new phase for 2nd DDS, 1st DDS DDS 0 degree phase
      else if(meas_mode == MODE_SWEEP_PHASE)
      {
        if(newPhase != oldPhase)
        {
          setLaser1_I(0.0);
          disableDDS();
          DDSI_SetFreqPhase(startFrequency, 0);
          DDSQ_SetFreqPhase(startFrequency, newPhase);
          enableDDS();
          oldPhase = newPhase;
          setLaser1_I(laser_current);
        }
        meas_counter++;
        if(meas_counter == meas_points)
        {
           meas_counter = 0;
           newPhase = oldPhase + sweepStep;
        }
      }
    // focus or position coil current sweep: set new current
      else if((meas_mode == MODE_SWEEP_FOCUS)
       || (meas_mode == MODE_SWEEP_POSITION))
      {
        if(newCurrent != oldCurrent)
        {
          if(meas_mode == MODE_SWEEP_FOCUS)
            setCoil1_I(newCurrent);
          if(meas_mode == MODE_SWEEP_POSITION)
            setCoil2_I(newCurrent);

          oldCurrent = newCurrent;
        }
        meas_counter++;
        if(meas_counter == meas_points)
        {
           meas_counter = 0;
          newCurrent = oldCurrent + sweepStep;
        }
      }
    // single frequency mode: set new frequency, only the first time
      else if(meas_mode == MODE_SINGLE_FREQ)
      {
        if(newFrequency != oldFrequency)
        {
          setLaser1_I(0.0);
          DDS_SetFreqInQuadrature(newFrequency);
          oldFrequency = newFrequency;
          setLaser1_I(laser_current);
        }
        meas_counter++;
      }
      break;
    }
    case 7:
    {
      setChannelGain(0, 1);      // trigger measurement on ADC0 with gain = 1
      break;
    }
    case 8:
    {
      volt_inphase = readVoltage();    // read-out voltage of ADC0
      setChannelGain(1, 1);      // trigger measurement on ADC1 with gain = 1
      break;
    }
    case 9:
    {
      volt_quadrature = readVoltage();  // read-out voltage of ADC1
      unsigned long temp_time = millis() - startTime; // get current time step
    // selection which parameter to write on Serial port
      if((meas_mode == MODE_SWEEP_FREQ) || (meas_mode == MODE_SINGLE_FREQ))
      {
        Serial.print(oldFrequency, 2);
      }
      if((meas_mode == MODE_SWEEP_FREQ_LOCKED)
    || (meas_mode == MODE_SWEEP_FREQ_LOCKED_DECADIC))
      {
        Serial.print(oldFrequency, 2);
      }
      if(meas_mode == MODE_SWEEP_PHASE)
      {
        Serial.print(oldPhase, 2);
      }
      if((meas_mode == MODE_SWEEP_FOCUS) || (meas_mode == MODE_SWEEP_POSITION))
      {
        Serial.print(oldCurrent, 2);
      }
    // send results over Serial port
      Serial.print("; "); Serial.print(temp_time, DEC);
      Serial.print("; "); Serial.print(volt_inphase, 9); Serial.print("; ");
      Serial.print(volt_quadrature, 9); Serial.print(";"); Serial.print("\n");

      break;
    }
  }
  // increment or reset the routine counter
  if(time_counter <= 9)
    time_counter++;
  else
```

```
    time_counter = 0;
}

void setup()
{
  Serial.begin(115200);

  timerID_meas = timer.setInterval(1L, measurementEvent);
  timer.disable(timerID_meas);

  setupHardware();
  Serial.println("Hello Alexander, welcome to project Laser-LIA");
}

// Restarts program but does not reset the peripherals and registers
void software_Reset()
{
  asm volatile ("  jmp 0");
}

void loop()
{
  timer.run();
}

/*
  SerialEvent occurs whenever a new data comes in the hardware serial RX. This
  routine is run between each time loop() runs. Multiple bytes of data may be
  available.
*/
char inputString[200];            // a String to hold incoming data
boolean stringComplete = false;   // whether the string is complete
int strCounter = 0;

void serialEvent()
{
  while (Serial.available())
  {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
    inputString[strCounter] = tolower(inChar);
    // if the incoming character is a newline, set a flag so the main loop can
    // do something about it:
    if ((inChar == '\n') || (strCounter == 199))
    {
      stringComplete = true;
      inputString[strCounter] = 0;
      strCounter = 0;
      if(parseCommand() == false)
      {
        Serial.print("Command not valid!\n");
      }
      inputString[0] = 0;
    }
    else
    {
      strCounter++;
    }
  }
}

```

```
// example: MODE SWEEP -f 32 33 0.001 kHz 2x 40mA
// example: MODE LOCKEDSWEEP -f 32.5 32 33 0.001 kHz 2x 40mA
// example: MODE SINGLE -f 32 kHz 100x 40mA
// example: MODE SWEEP -p 32 kHz 0 90 0.1 degree 2x 40mA
// example: STOP
bool parseCommand()
{
  char cmpStr[10];
  //Serial.println(inputString);

  char f1[10];
  char f2[10];
  char f3[10];
  char f4[10];
  // Frequency Sweep Mode
  if(sscanf(inputString, "mode sweep -f %9s %9s %9s %9s %dx %dma", f1, f2, f3,
      cmpStr, &meas_points, &laser_current) == 6)
  {
    meas_mode = MODE_SWEEP_FREQ;
    startFrequency = atof(f1);
    stopFrequency = atof(f2);
    sweepStep = atof(f3);
    if(strcmp(cmpStr, "khz") == 0)
    {
      startFrequency *= 1000;
      stopFrequency *= 1000;
      sweepStep *= 1000;
    }
    oldFrequency = -1.0;              // in Hz
    newFrequency = startFrequency;  // in Hz
    meas_counter = 0;
    time_counter = 0;

    Serial.print("Begin frequency sweep: "); Serial.print(startFrequency, 2);
  Serial.print("; "); Serial.print(stopFrequency, 2); Serial.print("; ");
    Serial.print(sweepStep, 2); Serial.print(" Hz; ");
    Serial.print(meas_points, DEC); Serial.print("x @ ");
    Serial.print(laser_current, DEC); Serial.print("mA\n");
    Serial.println(str1);

    timer.enable(timerID_meas);
    startTime = millis();
    return true;
  }
  // Locked Frequency Sweep Mode
  else if(sscanf(inputString, "mode lockedsweep -f %9s %9s %9s %9s %9s %dx %dma",
      f1, f2, f3, f4, cmpStr, &meas_points, &laser_current) == 7)
  {
    meas_mode = MODE_SWEEP_FREQ_LOCKED;
    lockedFrequency = atof(f1);
    startFrequency = atof(f2);
    stopFrequency = atof(f3);
    sweepStep = atof(f4);
    if(strcmp(cmpStr, "khz") == 0)
    {
      lockedFrequency *= 1000;
      startFrequency *= 1000;
      stopFrequency *= 1000;
      sweepStep *= 1000;
    }
    oldFrequency = -1.0;              // in Hz
    newFrequency = startFrequency;  // in Hz
    meas_counter = 0;
    time_counter = 0;

    Serial.print("Begin locked frequency sweep: ");
  Serial.print(lockedFrequency, 2); Serial.print("; ");
    Serial.print(startFrequency, 2); Serial.print("; ");
    Serial.print(stopFrequency, 2); Serial.print("; ");
    Serial.print(sweepStep, 2); Serial.print(" Hz; ");
    Serial.print(meas_points, DEC); Serial.print("x @ ");
    Serial.print(laser_current, DEC); Serial.print("mA\n");
    Serial.println(str1);

    timer.enable(timerID_meas);
    startTime = millis();
    return true;
  }
  // Decadic Locked Frequency Sweep Mode
  else if(sscanf(inputString, "mode lockedsweep -d %9s %9s %9s %9s %9s %dx %dma",
      f1, f2, f3, f4, cmpStr, &meas_points, &laser_current) == 7)
  {
    meas_mode = MODE_SWEEP_FREQ_LOCKED_DECADIC;
    lockedFrequency = atof(f1);
    startFrequency = atof(f2);
    stopFrequency = atof(f3);
    sweepStep = atof(f4);
    if(strcmp(cmpStr, "khz") == 0)
    {
      lockedFrequency *= 1000;
      startFrequency *= 1000;
      stopFrequency *= 1000;
      sweepStep *= 1000;
    }
    oldFrequency = -1.0;              // in Hz
    newFrequency = startFrequency;  // in Hz
    meas_counter = 0;
    time_counter = 0;

    Serial.print("Begin locked frequency sweep: ");
  Serial.print(lockedFrequency, 2); Serial.print("; ");
    Serial.print(startFrequency, 2); Serial.print("; ");
    Serial.print(stopFrequency, 2); Serial.print("; ");
    Serial.print(sweepStep, 2); Serial.print(" Hz; ");
    Serial.print(meas_points, DEC); Serial.print("x @ ");
    Serial.print(laser_current, DEC); Serial.print("mA\n");
    Serial.println(str1);

    timer.enable(timerID_meas);
    startTime = millis();
    return true;
  }
  // Phase Sweep Mode
  else if(sscanf(inputString, "mode sweep -p %9s %9s %9s %9s %9s degree %dx %dma",
         f1, cmpStr, f2, f3, f4, &meas_points, &laser_current) == 7)
  {
    meas_mode = MODE_SWEEP_PHASE;
    startFrequency = atof(f1);
    startPhase = atof(f2);
    stopPhase = atof(f3);
    sweepStep = atof(f4);
    if(strcmp(cmpStr, "khz") == 0)
    {
      startFrequency *= 1000;
    }
    newFrequency = startFrequency;
    oldPhase = -1.0;              // in degree
    newPhase = startPhase;        // in degree
    meas_counter = 0;
    time_counter = 0;

    Serial.print("Begin phase sweep: "); Serial.print(startFrequency, 2);
  Serial.print(" Hz, "); Serial.print(startPhase, 2); Serial.print("; ");
    Serial.print(stopPhase, 2); Serial.print("; ");
    Serial.print(sweepStep, 2); Serial.print(" degree; ");
    Serial.print(meas_points, DEC); Serial.print("x @ ");
    Serial.print(laser_current, DEC); Serial.print("mA\n");
    Serial.println(str2);

    timer.enable(timerID_meas);
    startTime = millis();
    return true;
  }
  // Foucs Sweep Mode
  else if(sscanf(inputString, "mode sweep -fc %9s %9s %9s %9s %9s ma %dx %dma",
         f1, cmpStr, f2, f3, f4, &meas_points, &laser_current) == 7)
  {
    meas_mode = MODE_SWEEP_FOCUS;
    startFrequency = atof(f1);
    startCurrent = atof(f2);
    stopCurrent = atof(f3);
    sweepStep = atof(f4);
    if(strcmp(cmpStr, "khz") == 0)
    {
      startFrequency *= 1000;
    }
    newFrequency = startFrequency;
    oldCurrent = -1.0;              // in mA
    newCurrent = startCurrent;      // in mA
    meas_counter = 0;
    time_counter = 0;

    Serial.print("Begin focus sweep: "); Serial.print(startFrequency, 2);
  Serial.print(" Hz, "); Serial.print(startCurrent, 2); Serial.print("; ");
    Serial.print(stopCurrent, 2); Serial.print("; ");
    Serial.print(sweepStep, 2); Serial.print(" mA; ");
    Serial.print(meas_points, DEC); Serial.print("x @ ");
    Serial.print(laser_current, DEC); Serial.print("mA\n");
    Serial.println(str3);

    timer.enable(timerID_meas);
    startTime = millis();
    return true;
  }
  // X-Position Sweep Mode
  else if(sscanf(inputString, "mode sweep -pc %9s %9s %9s %9s %9s ma %dx %dma",
         f1, cmpStr, f2, f3, f4, &meas_points, &laser_current) == 7)
  {
    meas_mode = MODE_SWEEP_POSITION;
    startFrequency = atof(f1);
    startCurrent = atof(f2);
    stopCurrent = atof(f3);
    sweepStep = atof(f4);
    if(strcmp(cmpStr, "khz") == 0)
    {
      startFrequency *= 1000;
    }
    newFrequency = startFrequency;
    oldCurrent = -1.0;                  // in mA
    newCurrent = startCurrent;        // in mA
    meas_counter = 0;
    time_counter = 0;

    Serial.print("Begin x-position sweep: "); Serial.print(startFrequency, 2);
  Serial.print(" Hz, "); Serial.print(startCurrent, 2); Serial.print("; ");
    Serial.print(stopCurrent, 2); Serial.print("; ");
    Serial.print(sweepStep, 2); Serial.print(" mA; ");
    Serial.print(meas_points, DEC); Serial.print("x @ ");
    Serial.print(laser_current, DEC); Serial.print("mA\n");
    Serial.println(str4);

    timer.enable(timerID_meas);
    startTime = millis();
    return true;
  }
  // stop modes
  else if(strcmp(inputString, "stop") == 0)
  {
    Serial.print("measurement stopped!");
    meas_mode = MODE_NO;
    timer.disable(timerID_meas);
    return true;
  }
  // single frequency mode
  else if(sscanf(inputString, "mode single -f %9s %9s %dx %dma",
          f1, cmpStr, &meas_points, &laser_current) == 4)
  {
    meas_mode = MODE_SINGLE_FREQ;
    startFrequency = atof(f1);
    if(strcmp(cmpStr, "khz") == 0)
    {
      startFrequency *= 1000;
    }
    oldFrequency = -1.0;                // in Hz
    newFrequency = startFrequency;    // in Hz
    meas_counter = 0;
    time_counter = 0;

    Serial.print("Begin single frequency: "); Serial.print(startFrequency, 2);
  Serial.print("; "); Serial.print(meas_points, DEC); Serial.print("x @ ");
    Serial.print(laser_current, DEC); Serial.print("mA\n");
    Serial.println(str1);

    timer.enable(timerID_meas);
    startTime = millis();
    return true;
  }
  else if(sscanf(inputString, "mode single -f %9s %9s %dmin %dma",
          f1, cmpStr, &meas_points, &laser_current) == 4)
  {
    meas_mode = MODE_SINGLE_FREQ;
    startFrequency = atof(f1);
    if(strcmp(cmpStr, "khz") == 0)
    {
      startFrequency *= 1000;
    }
    oldFrequency = -1.0;                // in Hz
    newFrequency = startFrequency;    // in Hz
    meas_counter = 0;
    time_counter = 0;
    meas_points = meas_points*60*100;

    Serial.print("Begin single frequency: "); Serial.print(startFrequency, 2);
  Serial.print("; "); Serial.print(meas_points, DEC); Serial.print("x @ ");
    Serial.print(laser_current, DEC); Serial.print("mA\n");
    Serial.println(str1);

    timer.enable(timerID_meas);
    startTime = millis();
    return true;
  }
  // no mode
  else if(strcmp(inputString, "reset") == 0)
  {
    meas_mode = MODE_NO;
    timer.disable(timerID_meas);
    software_Reset();
  }
  else
  {
    //meas_mode = MODE_NO;
    return false;
  }

}
```

E MATLAB Communication Script

```matlab
clear all
close all
clc

%% Measurement parameters ---------------------------------------------------

COMport = 3;
mode = 1;                % mode = 1: frequency sweep
                         % mode = 2: phase sweep
                         % mode = 3: single frequency n-times
                         % mode = 4: single frequency in min
                         % mode = 5: locked frequency sweep
                         % mode = 6: decadic locked frequency sweep

% frequencies in Hz
freqstart = 32750;
freqstop = 32850;
freqstep = 0.1;
freqsingle = 32815.5;
freqlocked = 32815.5;

% phase in degree
phasestart = 0;
phasestop = 360;
phasestep = 0.25;

% laser current in mA
current = 80;

% measurement points
points = 100;
min = 2;

modestr1 = ['MODE SWEEP -f ', num2str(freqstart), ' ', num2str(freqstop), ' ', num2str(freqstep), ' Hz ', num2str(points), 'x ', num2str(current), 'mA'];

modestr2 = ['MODE SWEEP -p ', num2str(freqsingle), ' Hz ', num2str(phasestart), ' ', num2str(phasestop), ' ', num2str(phasestep), ' degree ', num2str(points), 'x ', num2str(current), 'mA'];

modestr3 = ['MODE SINGLE -f ', num2str(freqsingle), ' Hz ', num2str(points) 'x ', num2str(current), 'mA'];
modestr4 = ['MODE SINGLE -f ', num2str(freqsingle), ' Hz ', num2str(min), 'min ', num2str(current), 'mA'];

modestr5 = ['MODE LOCKEDSWEEP -f ', num2str(freqlocked), ' ', num2str(freqstart), ' ', num2str(freqstop), ' ', num2str(freqstep), ' Hz ', num2str(points), 'x ', num2str(current), 'mA'];

modestr6 = ['MODE LOCKEDSWEEP -d ', num2str(freqlocked), ' ', num2str(freqstart), ' ', num2str(freqstop), ' ', num2str(freqstep), ' Hz ', num2str(points), 'x ', num2str(current), 'mA'];

resetcmd = 'reset';
teststr1 = 'MODE SWEEP -f 32 33 0.1 kHz 1x 80mA';
teststr2 = 'MODE SWEEP -p 32 kHz 0 90 1 degree 1x 40mA';
teststr3 = 'MODE SINGLE -f 32 kHz 100x 40mA';
teststr5 = 'MODE LOCKEDSWEEP -f 32.5 32 33 0.001 kHz 2x 40mA';
endcmd = ['measurement done!',char(10)];
```

```

%% Arduino connect and read-In ------------------------------------------

if(mode == 1)
    datalen = points * (freqstop - freqstart) / freqstep;
    usestr = modestr1;
elseif(mode == 2)
    datalen = points * (phasestop - phasestart) / phasestep;
    usestr = modestr2;
elseif(mode == 3)
    datalen = points;
    usestr = modestr3;
elseif(mode == 4)
    datalen = min*60*100;
    usestr = modestr4;
elseif(mode == 5)
    datalen = points * (freqstop - freqstart) / freqstep;
    usestr = modestr6;
elseif(mode == 6)
    datalen = points * ceil(log(freqstop/freqstart)) / freqstep;
    usestr = modestr7;
end

n = 1;
data=cell(1,datalen);
A = zeros(datalen,4);

uno = serial(['COM', num2str(COMport)], 'BaudRate', 115200);
fopen(uno)
fprintf(uno, resetcmd)
pause(5);
data-n  = fscanf(uno);
n = n + 1;

fprintf(uno, usestr)

endflag = 1;
while(endflag)
    temp = fscanf(uno)
    data-n  = temp;
    if(strcmp(data-n , endcmd))
        endflag = 0;
    else
        if(n ¿ 3)
            A(n-3,:) = sscanf(temp, '%f;', [1 4]);
        end
        n = n + 1;
    end
end

%% Shutdown Arduino -------------------------------------------------------

fprintf(uno, resetcmd)
fclose(uno)
delete(uno)
clear uno

A(A(:,1) == 0,:) = [];
time = A(:,2);
inphase = A(:,3);
quadrature = A(:,4);

```

```
%% Save results in file ---------------------------------------------------

time = datestr(now);
time = strrep(time, ':','-');
time = strrep(time, ' ',' ');
if(mode == 1)
    filename = ['freqsweepmeas', time, '.txt'];
elseif(mode == 2)
    filename = ['phasesweepmeas', time, '.txt'];
elseif(mode == 3)
    filename = ['singlefreqmeas', time, '.txt'];
elseif(mode == 4)
    filename = ['singlefreqmeas', time, '.txt'];
elseif(mode == 5)
    filename = ['lockedfreqsweepmeas', time, '.txt'];
elseif(mode == 6)
    filename = ['lockedfreqsweepmeas', time, '.txt'];
end

fileID = fopen(filename,'w');
fprintf(fileID,data-3);
fprintf(fileID,'%f; %f; %f; %f“n', A');
fclose(fileID);

%% Plot with time base ----------------------------------------------------

figure
plot(time./1000, inphase)
hold on
plot(time./1000, quadrature)
xlabel('Time [s]')
ylabel('Voltage [V]')
legend('V-inphase','V-quadrature')

if(mode == 1)           %frequency sweep
    freq = A(:,1);
    figure
    plot(freq./1000, inphase)
    hold on
    plot(freq./1000, quadrature)
    xlabel('Frequency [kHz]')
    ylabel('Voltage [V]')
    legend('V-inphase','V-quadrature')
elseif(mode == 2)       %phase sweep
    phase = A(:,1);
    figure
    plot(phase, inphase)
    hold on
    plot(phase, quadrature)
    xlabel('Phase [degree]')
    ylabel('Voltage [V]')
    legend('V-inphase','V-quadrature')
elseif((mode == 3) —— (mode == 4))      %single frequency
    freq = A(:,1);
    figure
    plot(time./1000, inphase)
    hold on
    plot(time./1000, quadrature)
    xlabel('Time [s]')
    ylabel('Voltage [V]')
    legend('V-inphase','V-quadrature')
elseif(mode == 5)
```

```
    freq = A(:,1);
    figure
    plot(freq./1000, inphase)
    hold on
    plot(freq./1000, quadrature)
    xlabel('Frequency [kHz]')
    ylabel('Voltage [V]')
    legend(' V -inphase ',' V -quadrature ')
elseif(mode == 6)
    freq = A(:,1);
    figure
    plot(freq./1000, inphase)
    hold on
    plot(freq./1000, quadrature)
    xlabel('Frequency [kHz]')
    ylabel('Voltage [V]')
    legend(' V -inphase ',' V -quadrature ')
end

%% Averaging and plot -----------------------------------------------------
avgpoints = points;

Mtemp = reshape(inphase,avgpoints,(length(inphase)/avgpoints));
avgtemp = (mean(Mtemp,1));
inphase = avgtemp';

Mtemp = reshape(quadrature,avgpoints,(length(quadrature)/avgpoints));
avgtemp = (mean(Mtemp,1));
quadrature = avgtemp';

if(mode == 1)          %frequency sweep
    freq = A(:,1);
    freq = freq(1:avgpoints:end);
    figure
    plot(freq./1000, inphase)
    hold on
    plot(freq./1000, quadrature)
    xlabel('Frequency [kHz]')
    ylabel('Voltage [V]')
    legend(' V -inphase ',' V -quadrature ')

    realpart = sqrt(inphase.^2 + quadrature.^2);
    imagpart = atan(quadrature./inphase).*180./pi;
    figure
    subplot(2, 1, 1)
    plot(freq./1000, realpart)
    xlabel('Frequency [kHz]')
    ylabel('Normalized Real Part')
    title('Frequency Sweep - Real')
    %xlim([32.73 32.77])
    subplot(2, 1, 2)
    plot(freq./1000, imagpart)
    xlabel('Frequency [kHz]')
    ylabel('Real and Imaginary')
    title('Frequency Sweep - Imag')
    %xlim([32.73 32.77])
elseif(mode == 2)      %phase sweep
    phase = A(:,1);
    phase = phase(1:avgpoints:end);
    figure
    plot(phase, inphase)
    hold on
    plot(phase, quadrature)
    xlabel('Phase [degree]')
    ylabel('Voltage [V]')
    legend(' V -inphase ',' V -quadrature ')
elseif((mode == 3) —— (mode == 4))      %single frequency
    time = A(:,2);
    time = time(1:avgpoints:end);
    figure
    plot(time./1000, inphase)
    hold on
    plot(time./1000, quadrature)
    xlabel('Time [s]')
    ylabel('Voltage [V]')
    legend(' V -inphase ',' V -quadrature ')

    realpart = sqrt(inphase. 2 + quadrature. 2);
    imagpart = atan(quadrature./inphase).*180./pi;
    figure
    subplot(2, 1, 1)
    plot(time./1000, realpart)
    xlabel('Frequency [kHz]')
    ylabel('Normalized Real Part')
    title('Frequency Sweep - Real')
    %xlim([32.73 32.77])
    subplot(2, 1, 2)
    plot(time./1000, imagpart)
    xlabel('Frequency [kHz]')
    ylabel('Real and Imaginary')
    title('Frequency Sweep - Imag')
    %xlim([32.73 32.77])
elseif(mode == 5)           %frequency sweep
    freq = A(:,1);
    freq = freq(1:avgpoints:end);
    figure
    plot(freq./1000, inphase)
    hold on
    plot(freq./1000, quadrature)
    xlabel('Frequency [kHz]')
    ylabel('Voltage [V]')
    legend(' V -inphase ',' V -quadrature ')
elseif(mode == 6)           %frequency sweep
    freq = A(:,1);
    freq = freq(1:avgpoints:end);
    figure
    plot(freq./1000, inphase)
    hold on
    plot(freq./1000, quadrature)
    xlabel('Frequency [kHz]')
    ylabel('Voltage [V]')
    legend(' V -inphase ',' V -quadrature ')
end
```