

MASTERARBEIT

Entwicklung und Aufbau
einer Laserüberwachung für ein Rayleigh-Mie-Raman
LIDAR

von

Liu Ning

(Studiengang: Elektrotechnik)

Gutachter: Prof. Dr. Nils Damaschke (Universität Rostock)

Dr. Michael Gerding (Leibniz-Institut für Atmosphärenphysik e.V.)

Rostock, 20.September 2011

Leibniz-Institut für Atmosphärenphysik e.V. in Kühlungsborn

Mathematisch-Naturwissenschaftliche Fakultät

Institut für Allgemeine Elektrotechnik

Fakultät für Informatik und Elektrotechnik

Universität Rostock

Inhaltverzeichnis

1. Einleitung	1
1.1 Motivation	1
1.2 Ziel und Aufgabenstellung der Masterarbeit	1
1.3 Kapitelübersicht der Masterarbeit	2
2. Physikalische Grundlagen	3
2.1 Funktionsweise des RMR-Lidar Systems am IAP	3
2.1.1 Prinzip	3
2.1.2 Sendeeinheit	5
2.1.2.1 Funktionsweise eines Nd:YAG-Lasers	5
2.1.2.2 Seederlaser	5
2.1.2.3 Leistungslaser	6
2.1.3 Empfangseinheit	6
2.2 Erfasste Parameter des Messsystems	7
2.2.1 Spannungsparameter	7
2.2.2 Temperaturparameter	8
2.3 Zusammenhänge zwischen erfassten Parametern	9
3. Hardware-Entwurf	10
3.1 Platine für den Fotodiodenverstärker	10
3.1.1 Transimpedanzverstärker	10
3.1.1.1 Funktionsweise	11
3.1.1.2 Mögliche Arten des Rauschens	11
3.1.1.3 Bauteilauswahl	12
3.1.1.4 Schaltung des Transimpedanzverstärkers	16
3.1.2 Aktive Tiefpass-Filter	17
3.1.2.1 Funktionsweise	17
3.1.2.2 Bauteilauswahl	18
3.1.2.3 Simulation des Schaltungsverhaltens	18
3.1.3 Spannungsversorgung	23
3.1.3.1 Funktionsweise	23

3.1.3.2 BauteilAuswahl.....	23
3.1.3.3 Schaltung der Spannungsversorgung	23
3.1.4 Gesamtschaltplan.....	24
3.1.5 Fertigung dieser Platine	25
3.2 Platine für die Temperaturmessung	26
3.2.1 Funktionsweise.....	26
3.2.2 BauteilAuswahl	26
3.2.3 Schaltung der Temperaturmessung	28
3.2.4 Fertigung dieser Platine	29
3.3 Aufbau der Gehäuse und Netzteil	30
3.3.1 Gehäuse.....	30
3.3.2 Netzteil	31
3.4 ADAM Module.....	31
3.4.1 ADAM 4017 Modul.....	31
3.4.2 ADAM 4520I Modul	32
3.5 Gesamtplan des Messsystems	32
3.6 Hardware-Test	34
3.6.1 Platine für den Fotodiodenverstärker.....	34
3.6.2 Platine für die Temperaturmessung	35
 4. Software-Design	 36
4.1 Idee und Programmablaufplan.....	36
4.2 Benutzeroberflächen.....	37
4.2.1 Datenübertragung.....	37
4.2.2 Datenauslesung	38
4.2.3 Datenberechnung	38
4.2.4 Datenspeicherung	39
4.2.5 Grafikerzeugung	39
4.3 Software-Test	40
4.3.1 Fehlermeldung : „ zu wenig Arbeitsspeicher“	40
4.3.2 Fehlermeldung : „EConvertError-Exception“	40
 5. Signal-und Datenverarbeitung.....	 42
5.1 Idee und Programmablaufplan.....	42

5.2 Analyse der erfassten Parametern	43
5.2.1 Spannungsparameter.....	43
5.2.2 Temperaturparameter.....	47
5.3 Zusammenhänge zwischen verschiedenen Parametern	51
6. Zusammenfassung und Ausblick.....	51
 Anhang A.....	 53
A.1 Quellcode vom Programm ADAM.....	53
A.1.1 Quellcode der Datenübertragung.....	55
A.1.2 Quellcode der Datenauslesung und Datenberechnung.....	59
A.1.3 Quellcode der Datenspeicherung.....	62
A.2 Quellcode vom Programm Adamlog (Grafikerzeugung).....	64
A.3 Quellcode der M-Files mit MATLAB.....	67
 Anhang B.....	 70
 Abbildungsverzeichnis.....	 71
 Tabellenverzeichnis.....	 73
 Literaturverzeichnis	 74
 Danksagung.....	 77

1. Einleitung

1.1 Motivation

Am Leibniz-Institut für Atmosphärenphysik an der Universität Rostock in Kühlungsborn (IAP) werden u.a. zeitlich und räumlich hochaufgelöste Lidar-Messungen bei Tag und bei Nacht durchgeführt. Die für Lidar-Anwendungen verwendeten Laser müssen wegen der hohen Anforderungen an Leistung und Präzision in teilweise kritischen, technischen Grenzbereichen betrieben werden. Um Schäden an den Systemen zu verhindern und die Betriebsparameter optimal einzustellen, soll in dieser Arbeit ein Überwachungsmodul für die Laserkühlung und die wichtigsten Leistungsparameter entwickelt und aufgebaut werden.

1.2 Ziel und Aufgabenstellung der Masterarbeit

Ziel der Masterarbeit ist die Entwicklung und der anschließende Aufbau einer Laserparameterüberwachung des neuen Kühlungsborner Rayleigh-Mie-Raman (RMR) Lidars. Die Parameter sollen online dargestellt und ausgewertet werden. Bei Überschreiten voreingestellter Grenzen soll der Laser automatisch ausgeschaltet werden, um Schäden am System zu verhindern. Die Messwerte sollen mit reduzierter Rate protokolliert werden. Die Auswertung dieser Protokolle ermöglicht die Optimierung von Betriebsparametern wie der Raum- oder Kühlwassertemperatur. In einer späteren Ausbauhilfe soll der Laser automatisch ausgeschaltet werden, um Schäden am System zu verhindern.

Diese Arbeit umfasst folgende Schwerpunkte:

- Aufbau einer Überwachungselektronik auf Basis von ADAM-Feldbus-Modulen (für A/D Wandler von Laserparametern, Temperatur und Wasserflussmessung)
- Entwicklung und Fertigung einer Verstärkerplatine für ein Fotodiodensignal (Überwachung der Leistung eines Dauerstrichlasers) für Ausgangsspannungen von 0-5V zum Anschluss an einen ADAM-A/D-Wandler
- Dokumentation des Messaufbaus (Hard- und Software, Leiterplattenlayout etc.)
- Auslesen der überwachten Parameter abhängig von jeweiliger Aktualisierungsrate und Darstellung der Werte auf einem PC
- Protokollierung der Messwerte in einer ASCII-Datei und automatische Bereitstellung in einem Protokoll-Datenarchiv
- Erkennen dynamischer Beziehungen im System
- Auswertung der Messwerte in Hinblick auf eine Optimierung der Laserparameter, d.h. mögliche Verbesserung der thermischen oder optischen Stabilität des Lasersystems

1.3 Kapitelübersicht der Masterarbeit

Diese Masterarbeit gliedert sich in sechs Kapitel: im ersten Kapitel wird die Motivation, die Aufgabenstellung sowie der Aufbau der Masterarbeit dargestellt. Im 2. Kapitel werden relevante atmosphärenphysikalische Grundlagen sowie die Funktionsweise des Rayleigh-Mie – Raman Lidars am IAP in Kühlungsborn vorgestellt. Kapitel 3 erörtert den Hardware-Entwurf. Hier wird der Aufbau des Gehäuses, die Entwicklung der Verstärkerplatine für ein Fotodiodensignal, die Fertigung der Platine für die Temperaturmessung, der Anschluss an sogenannte ADAM Module (für A/D Wandler) sowie der Hardware Test. Im 4. Kapitel wird die Softwareentwicklung mit Delphi 6 vorgestellt (Ideenskizzen und Flussdiagramme). Kapitel 5 zeigt beispielhaft einige Messergebnisse sowie Ergebnisse der Signal-Datenverarbeitung. Im Kapitel 6 werden eine Zusammenfassung sowie ein Ausblick gegeben.

2. Physikalische Grundlagen

In diesem Kapitel wird das ab 2009 am IAP in Kühlungsborn in Betrieb genommene tageslichtfähige RMR-Lidar vorgestellt.

Aufgrund der Klimaänderungen spielt die Erforschung der Atmosphäre eine immer größere Rolle. Die Atmosphäre der Erde kann hinsichtlich der Temperaturstruktur von unten nach oben in die Schichten: Troposphäre, Stratosphäre, Mesosphäre und Thermosphäre eingeteilt werden. Wichtige Eigenschaften sind Temperatur, Dichte, Druck und chemische Zusammensetzung der Atmosphäre. Am IAP werden die thermische und dynamische Struktur der Atmosphäre (z.B. Temperatur-Struktur, Morphologie von Schwerewellen, jahreszeitliche Variation der Temperatur, Winde), Aerosole (z.B. Leuchtende Nachtwolken) sowie Spurengase in der mittleren Atmosphäre (z.B. Wasserdampf, Metallatome, Ozon) erforscht, wie in u.a. in [Institutsbericht 2008/2009] ausführlich dargestellt wird. Das RMR-Lidar misst davon vor allem die Temperaturschichten und ihre zeitliche Veränderung in der Stratosphäre und Mesosphäre.

2.1 Funktionsweise des RMR-Lidar Systems am IAP

2.1.1 Prinzip

Ein Lidar (englisch: **L**ight **d**etection and **r**anging) ist ein aktives optisches Messinstrument, bei dem aus der Atmosphäre rückgestreutes Laserlicht untersucht wird. Abbildung 2.1 zeigt das Grundprinzip eines RMR-Lidars am IAP in Kühlungsborn. Das vom Laser ausgesendete Licht wird in der Atmosphäre von den Molekülen der Luft und kleinen Teilchen gestreut. Es wird danach von einem Teleskopspiegel (Durchmesser von 80 cm) empfangen und in eine Glasfaser fokussiert. Die Photonen werden durch die Glasfaser zum Detektor übertragen. Die Intensität des gestreuten Signals ist grundsätzlich proportional zur Dichte, wenn es keine Aerosole gibt. Über die Laufzeitverzögerung des Lichtes liefert das empfangene Signal höhen aufgelöste Informationen über die Dichte der Atmosphäre. Durch Integration der hydrostatischen Grundgleichung (siehe Anhang B) kann daraus die Temperatur berechnet werden. Aus der Rückstreuung an Aerosolen können zusätzliche Informationen über den Zustand der Atmosphäre gewonnen werden. Am IAP gibt es verschiedene Lidarsysteme: u.a. ein Kalium-Resonanzlidar, sowie zwei RMR-Lidar-Systeme. Der Begriff „RMR-Lidar“ bedeutet, dass drei Messmethoden kombiniert werden. Dies sind **R**ayleigh-Streuung, **M**ie-Streuung und **R**aman-Streuung. Seit 2002 werden am Institut Nachtmessungen mit dem RMR-Lidar durchgeführt [Alpers et al. 2004].

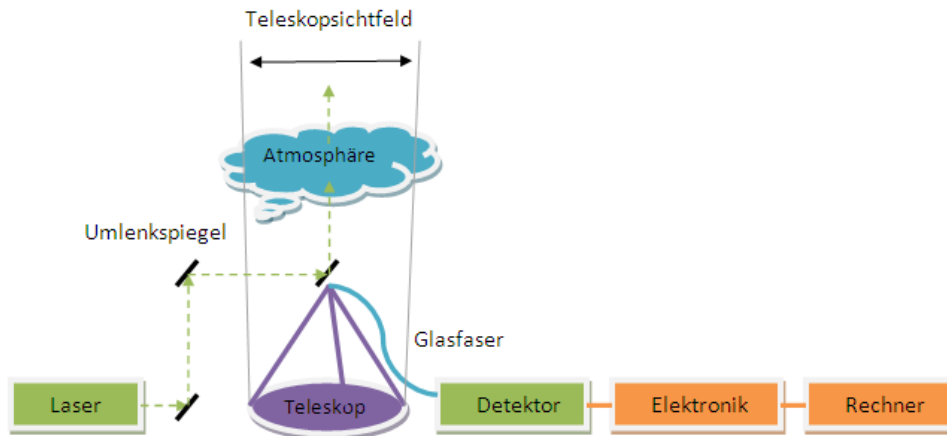


Abbildung 2.1 Lidarprinzipskizze

Abbildung 2.2 zeigt ein typisches Temperaturprofil des Nacht-RMR-Lidarsystems in Kombination mit dem Kalium-Resonanzlidar (Wellenlänge $\lambda=770$ nm) des Instituts. Durch die Kombination beider Systeme ist eine Temperatursondierung zwischen 2 und ca. 105 km möglich.

Zur Durchführung von tageszeitlich unabhängigen Messungen wurde 2009 ein neues RMR-Lidar aufgebaut, mit dem Messungen auch am Tag möglich sind. Das RMR-Lidar ist in seiner Leistungsfähigkeit abhängig von der Untergrundstrahlung der Sonne. Durch optische und räumliche Filtertechniken ist es beim neuen RMR-Lidar jedoch gelungen, das Signal-Rausch-Verhältnis so weit zu verbessern, dass auch Messungen bei vollem Tageslicht möglich werden. Für dieses neue RMR-Lidar soll eine Überwachung verschiedener Laserparameter aufgebaut werden.

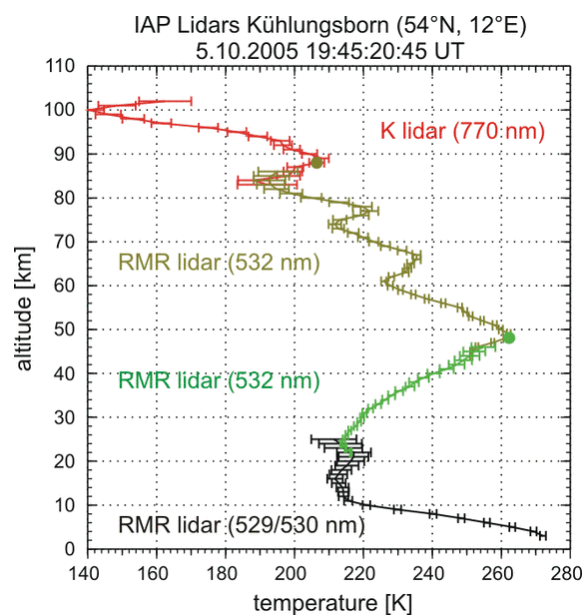


Abbildung 2.2 Temperaturprofil zwischen 1 und 105 km am 5.10.2005. Mittel über eine Stunde der Messung, K-Lidar und RMR-Lidar in Kühlungsborn [Institutsbericht 2006]

Im folgenden Überblick beschreibe ich die Sende- und Empfangseinheit des neuen RMR-Lidars.

2.1.2 Sendeeinheit

Die Sendeeinheit des RMR-Lidars in Kühlungsborn besteht im Wesentlichen aus einem blitzlampengepumpten Leistungslaser (Nd:YAG-Laser), der über einen diodengepumpten Nd:YAG Seederlaser in seiner Frequenz festgelegt wird. Darüber hinaus wird das Licht über ein Aufweitungsteleskop und mehrere Umlenkspiegel in die Atmosphäre gesendet.

2.1.2.1 Funktionsweise eines Nd:YAG-Lasers

Nd:YAG-Laser sind Festkörperlaser, bei denen Laserübergänge eines Nd-dotierten Yttrium-Aluminium-Granats ausgenutzt werden. Abbildung 2.3 zeigt die für die Laserübergänge relevanten Energieniveaus eines Nd:YAG-Lasers, der ein 4-Niveaulaser ist. Die Anregung höherer Energiezustände erfolgt technisch durch optisches Pumpen mit Blitzlampen. Der Übergang in das obere Niveau $^4F_{3/2}$ erfolgt strahlungslos. Photonen der Wellenlänge von $\lambda=1064$ nm (infrarot) werden durch den Laserübergang bei der Abregung auf das Niveau $^4I_{1/2}$ emittiert. Durch Frequenzverdopplung in einem nichtlinearen optischen Medium wird die für Temperaturmessungen in der Atmosphäre empfindlichste Wellenlänge von 532 nm erzeugt.

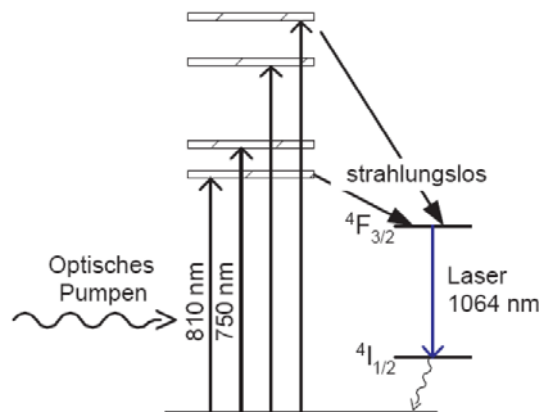


Abbildung 2.3 Termschema eines Nd:YAG-Lasers [Damaschke 2010]

2.1.2.2 Seederlaser

Der eingesetzte Seederlaser ist ein cw (englisch: continuous wave)-Nd:YAG Laser der Innolight GmbH mit der Fundamentalwellenlänge $\lambda=1064$ nm und der weiteren Wellenlänge $\lambda=532$ nm, die nach Frequenzverdopplung in einem nichtlinearen optischen Kristall erzeugt wird. Injection Seeding ist ein häufiges Verfahren, um gepulste Laser mit einer hohen Verstärkung im Monomodebetrieb zu halten. Die spektrale Stabilisierung des Seederlasers erfolgt auf einer Jod-Absorptionslinie bei der Wellenlänge von 532 nm [Söder 2010]. Die Ausgangsleistung beträgt 1,05 W bei $\lambda=1064$ nm und 26 mW bei $\lambda=532$ nm.

2.1.2.3 Leistungslaser

Der Leistungslaser ist ein gepulster Nd:YAG Laser vom Typ Spectra Physics PRO290-30 der Quanta-Ray Nd:YAG Laser-Familie mit einer Pulsrate von 30 Hz. Die Pulsenergie bei $\lambda=1064$ nm beträgt 800 mJ, bei $\lambda=532$ nm 500 mJ im geseedeten Betrieb.

2.1.3 Empfangseinheit

Die Empfangseinheit des RMR-Lidars in Kühlungsborn besteht aus einem Teleskop und einer Nachweisbank. Das aus der Atmosphäre zurückgestreute Licht wird zuerst von dem Teleskop erfasst (Parabolspiegel mit Durchmesser von 80 cm) und wird auf den Kern einer Multimodeglasfaser (200 μ m Durchmesser) fokussiert. Anschließend wird das Licht durch die Glasfaser in die Nachweisbank geleitet. Ein schematischer Aufbau des Nachweiszweiges des tageslichtfähigen RMR-Lidars ist in Abbildung 2.4 dargestellt.

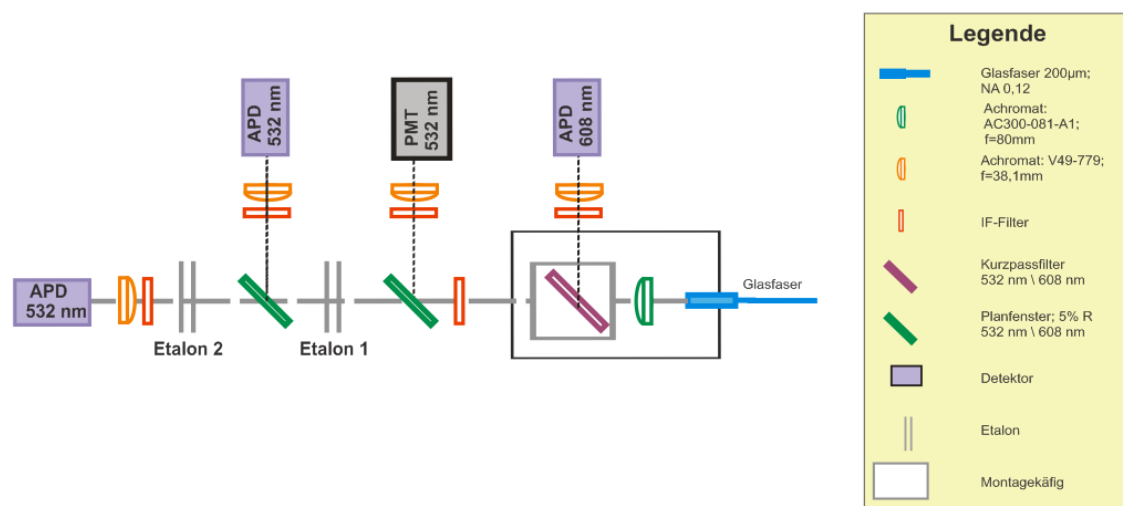


Abbildung 2.4 Schematischer Aufbau des Nachweiszweiges des tageslichtfähigen RMR-Lidars [Institutsbericht IAP 2010]

Mit dem neuen tageslichtfähigen RMR-Lidar können bei Tag Temperaturen sowie sogenannte „Leuchtende Nachtwolken“ (Eiswolken in ca. 83 km Höhe, die gelegentlich im Sommer auftreten) untersucht werden. Das Licht gelangt von rechts über eine Glasfaser in den Nachweiszweig und wird zunächst mit einem Achromaten parallelisiert. Mit Hilfe dichroitischer Strahlteiler wird das Licht nach Rayleigh/Mie- und Raman- Wellenlängen selektiert. Die Photonen mit einer Wellenlänge von etwa 532 nm gelangen danach auf einen Interferenzfilter mit einer Halbwertsbreite von 130 pm, der bereits einen Großteil der unerwünschten Untergrundstrahlung herausfiltert. Desweiteren erfolgt eine spektrale Filterung mit zwei Fabry-Perot-Etalons. Das erste hat eine Halbwertsbreite von 4 pm und einen freien Spektralbereich von 120 pm, das zweite ist nur unwesentlich breiter mit einer

Halbwertsbreite von 4,5 pm und einem freien Spektralbereich von 140 pm. Nach der spektralen Filterung wird das Licht mit Achromaten auf die Fläche des jeweiligen Detektors (Avalanche-Photo-Dioden, APDs bzw. Photomultiplier, PMT) fokussiert. Der Photomultiplier dient zur Kontrolle der Etalontransmission. Mit Hilfe der empfindlichen Detektoren (APDs) wird das Licht im Einzelphotonenbetrieb detektiert.

2.2 Erfasste Parameter des Messsystems

Im Rahmen meiner Masterarbeit wurde die Messung von bislang sieben Parametern durchgeführt. Diese lassen sich in zwei Gruppen unterteilen: die Spannungsgruppe und die Temperaturgruppe. Durch die Überwachung dieser Parameter kann festgelegt werden, wie gut die Regelung läuft.

2.2.1 Spannungsparameter

1. BUTR (Build Up Time Reduction)-Spannung

Die sogenannte „BUTR“ (englisch: **B**uild **U**p **T**ime **R**eduction) ist der zeitliche Unterschied zwischen der Q-Switch-Emission im „Seeding“-Betrieb zum „ungeseedeten Betrieb“. Mit der BUTR wird die Güte des Seedings bestimmt. Die Regelung des Laserresonators und des Seederlaser wird über dieses BUTR-Verfahren realisiert. Eine optimale BUTR-Spannung kann über Seederlaser eine optimale Frequenzstabilität bekommen. Zur Kontrolle kann eine der BUTR proportionale Spannung an einer BNC (englisch: **B**ayonet **N**eill **C**oncelman)-Schnittstelle abgegriffen werden.

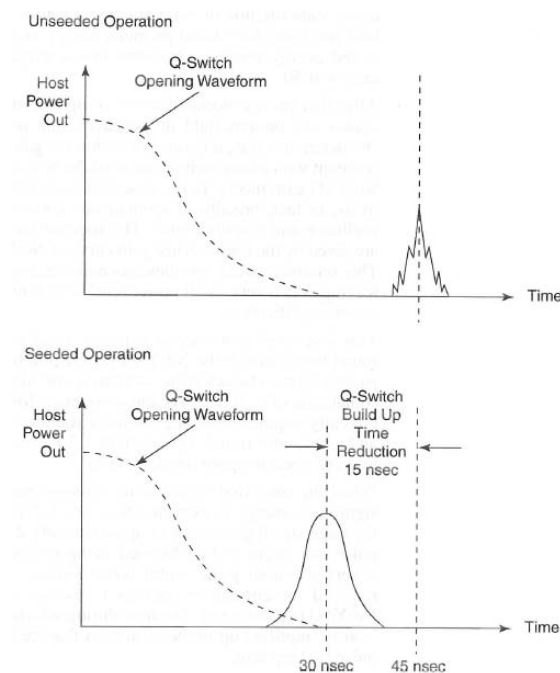


Abbildung 2.5 Q-Switch Build-up-time Reduction [Spectra Physics Injection Seeder]

Abbildung 2.5 zeigt das Grundprinzip der Build-up-time Reduction. Der zeitliche Unterschied ist 15 ns zwischen der Q-Switch-Emission im „Seeding“-Betrieb (unteres Schaubild) zum „ungeseedeten Betrieb“ (oberes Schaubild).

2. Piezo-Spannung

Der piezogesteuerte Endspiegel im Laserresonator regelt die Länge des Laserresonators. Auf das Piezoelement kann die Länge des Laserresonators bestimmt und herangezogen werden. Diese Piezo-Spannung ändert sich aufgrund thermischer Ausdehnung der Bauteile und muss permanent angepasst werden. Diese Piezo-Spannungsdifferenz kann in eine Wellenlängenänderung umgerechnet werden, d.h. durch diese Parameter kann das Wellenlängenstabilisierungsniveau bestimmt werden. Zur Kontrolle kann eine der Piezo-Spannung proportionale Spannung an einer BNC-Schnittstelle abgegriffen werden.

3. Fotodioden-Spannung 1

Mit Hilfe einer Fotodiode wird die Leistung des Seederlasers bei $\lambda=532$ nm gemessen. Sie dient der Überwachung des Seederlasers und den dortigen Frequenzverdopplung.

4. Fotodioden-Spannung 2

Zudem wird mit Hilfe einer Fotodiode misst die Leistung des Seederlasers bei $\lambda= 1064$ nm. Sie dient ebenso zur Überwachung des Seederlasers.

2.2.2 Temperaturparameter

Die Kontrolle der Temperatur spielt während des Laserbetriebes eine große Rolle. Daher ist es wichtig, das Lasersystem zu stabilisieren. In der Arbeit sollen drei Temperaturen, die Laser-Kopftemperatur, die Raumtemperatur sowie die Kühlwassertemperatur gemessen werden.

1. Laser-Kopftemperatur

Die Laserlichtquelle besteht aus einem Laserkopf, einer Spannungsversorgung und einer Kühlung. In dem Laserkopf befinden sich u.a. die Blitzlampen und die Laserstäbe. Um eine Überhitzung des Laserkopfes zu vermeiden, ist eine Temperaturüberwachung notwendig.

2. Raumtemperatur

Das Lasersystem soll in einer stabilisierten Atmosphäre betrieben werden. Deswegen wird eine Klimaanlage im Labor eingesetzt, um die Temperatur zu stabilisieren. Die Raumtemperaturüberwachung dient der Kontrolle der Klimaanlage, die insbesondere im Sommer am Rand ihrer Leistungsfähigkeit betrieben wird.

3. Kühlwassertemperatur

Aufgrund der hohen Anschlussleistung der Pumpquellen, wächst die Laser-Kopftemperatur immer weiter an. Die Laser-Kopftemperatur wird durch den Kühlwasserkreislauf reguliert, der im optimalen Temperaturbereich gehalten werden kann. Der Kühlwasserfluss muss vor

dem Bedienen eingeschaltet werden. Ein NTC100 (Heißleiter) als Temperatursensor ist auf dem Wasserhahn aufgebracht. Durch die Messung kann die Kühlwassertemperatur überwacht und z.B. ein fehlender Kühlwasserfluss erkannt werden.

2.3 Zusammenhänge zwischen erfassten Parametern

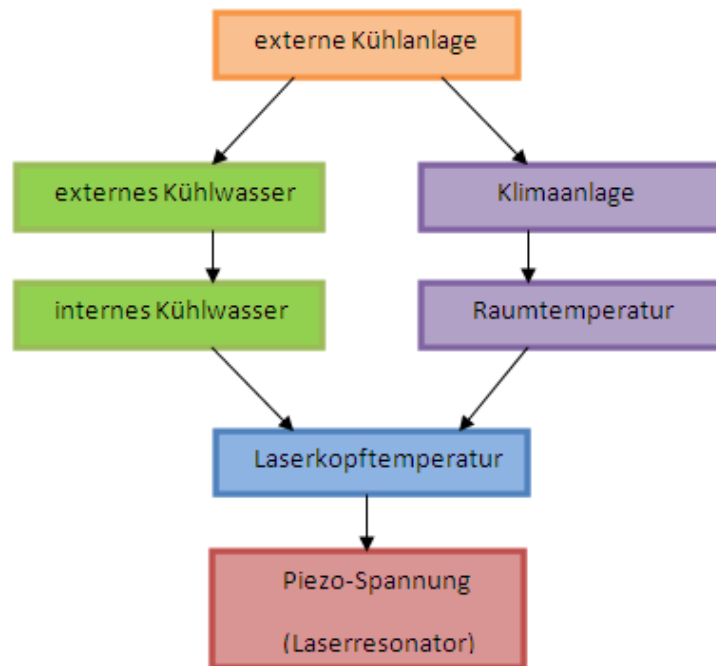


Abbildung 2.6 Zusammenhänge zwischen erfassten Parametern

Abbildung 2.6 zeigt die Zusammenhänge zwischen erfassten Parametern. Es gibt zuerst eine externe Kühlanlage, die das Kühlwasser für die Laser und die Klimaanlage versorgt. Die externe Kühlanlage bestimmt das externe Kühlwasser. Durch die Klimaanlage lässt sich die Raumtemperatur in einem bestimmten Temperaturbereich stabilisieren. Das externe Kühlwasser beeinflusst später das interne Kühlwasser. Das interne Kühlwasser bestimmt direkt die Laserkopf-Temperatur. Die Temperaturvariation im internen Kühlwasser verursacht Längenänderungen im Laserresonator. Die Längenänderung im Laserresonator wird durch die Piezo-Spannung ausgeglichen. Außerdem kann die Raumtemperatur schwach die Laserkopftemperatur beeinflussen.

3. Hardware-Entwurf

In diesem Kapitel wird das aufgebaute Gehäuse und die Auswahl der elektrischen Bauteile vorgestellt und das Prinzip der entwickelten Schaltung verdeutlicht. Zuerst werden hierzu die Anforderungen an die Bauelemente erläutert und dabei die wichtigsten Auswahlkriterien angegeben. Darüber hinaus werden einige notwendige Grundelemente der Schaltung erläutert.

Der Strukturplan des in dieser Arbeit entwickelten Messsystems ist in Abbildung 3.1 dargestellt.

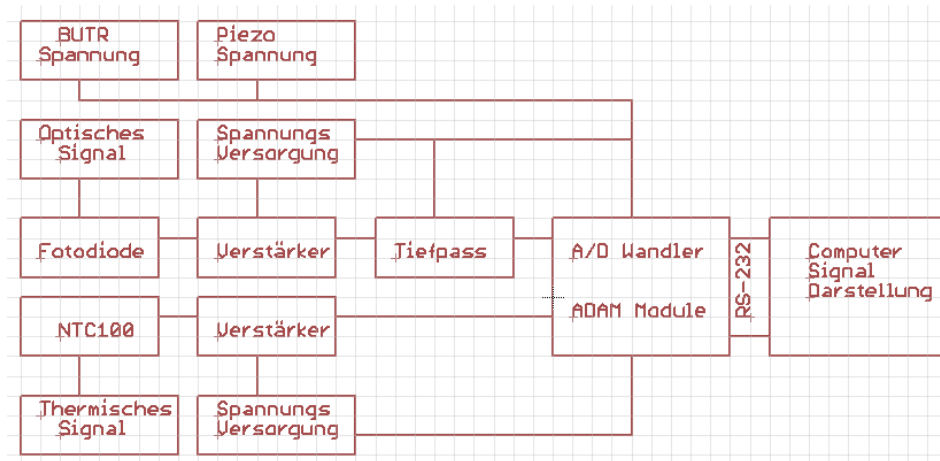


Abbildung 3.1 Strukturplan des in dieser Arbeit entwickelten Messsystems

Optische und thermische Signale des Lasers werden von Sensoren (Fotodiode und NTC100) empfangen und in elektrische Signale umgewandelt. Diese Signale werden zur passenden Schaltung, wie einem Transimpedanzverstärker weitergeleitet, um dort eine elektrische Spannung messen zu können. Als A/D Wandler werden ADAM 4017 I/O-Module der Firma Advantech verwendet. In den Modulen werden die Signale verarbeitet und einem Rechner über RS-232 Schnittstelle zur Weiterarbeitung und Analyse zur Verfügung gestellt. Mit einem in Delphi 6 erstellten Programm können die Parameter des Lasersystems einfach kontrolliert und analysiert werden.

Zum Schluss wird ein Gesamtplan und zwei PCB (englisch: **P**rinted **C**ircuit **B**oard) Layouts gezeigt. Das EDA Programm Eagle wurde zur Erstellung der Layouts verwendet.

3.1 Platine für den Fotodiodenverstärker

3.1.1 Transimpedanzverstärker

Die Schaltung des Transimpedanzverstärkers soll die Signale der Fotodioden (hier: cw Laser 532 nm und 1064 nm) empfangen. Sie besteht aus der Empfangseinheit, welche die Wandlung der optischen in elektrische Signale vornimmt und die elektrischen Signale verstärkt. Darüber hinaus soll ein aktiver Tiefpassfilters das Rauschen reduzieren und somit ein besseres Signal

liefern. Um eine präzise Messung des Fotostroms zu haben, nutzt man einen Transimpedanzverstärker. Eine Fotodiode liefert einen Eingangsstrom proportional zum Lichteinfall. Um eine Auswertung dieser Lichtverstärkung zu bekommen, ist ein Signal mit entsprechender Amplitude nötig. Das Signal der Fotodiode muss sowohl verstärkt als auch in eine Spannung umwandelt werden.

3.1.1.1 Funktionsweise

Transimpedanzverstärker liefern am Ausgang eine Spannung, welche proportional zum Fotostrom ist. Deswegen nennt man den Transimpedanzverstärker auch Strom-Spannungswandler. Den prinzipiellen Aufbau zeigt Abbildung 3.2.

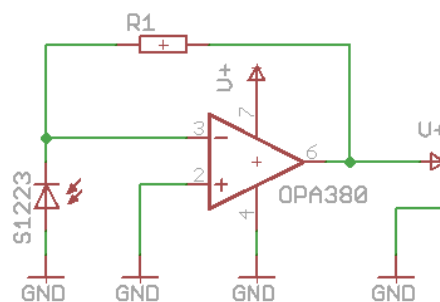


Abbildung 3.2 Prinzipieller Aufbau des Transimpedanzverstärkers mit Fotodiode

Wie Abbildung 3.2 zeigt, wird eine typische Schaltung eines Transimpedanzverstärkers aus einem Operationsverstärker und einem Rückkopplungswiderstand aufgebaut. Das Signal der Fotodiode wird an den invertierenden Eingang des Operationsverstärkers gelegt. Der Eingang ist also stromgesteuert. Der Ausgang verhält sich wie eine Spannungsquelle, wobei $U = I \cdot Z$ gilt. U ist die Ausgangsspannung, I ist der Eingangsstrom und Z ist die Transimpedanz.

Schaltungstechnisch handelt es sich bei der Transimpedanz Z um den Innenwiderstand des Transimpedanzknotens. Je höher Z ist, desto größer wird die Differenzverstärkung. Wenn in den Operationsverstärker-Eingang idealerweise kein Strom fließt, ergibt sich, dass Transimpedanz und Widerstand gegengleich sind, also $Z = -R$.

Aus den Überlegungen folgt, dass die Fotodiode mit der Anode an Masse und mit der Kathode an den Eingang angeschlossen wird, so dass bei Lichteinfall eine positive Ausgangsspannung anfällt. Der Widerstandswert sollte sehr groß sein, da das Signal der Fotodiode sehr klein ist.

3.1.1.2 Mögliche Arten des Rauschens

Das Rauschsignal spielt bei der Schaltung des Transimpedanzverstärkers eine große Rolle. Bei genügend großer Verstärkung tritt das Rauschen bereits auf, wenn noch kein Lichteinfall vorhanden ist. Das Rauschen hat prinzipielle physikalische Gründe und kann nie vollständig

eliminiert werden. Verschieden wichtige Arten des Rauschens, die sich später überlagern, können unterschieden werden [Müller, R1990].

Thermisches Rauschen

Wärmerauschen, das aus der thermischen Bewegung der Ladungsträger in elektrischen Schaltkreisen hervorgeht, ist ein so genanntes „weißes Rauschen“, also frequenzunabhängig. Der Effektivwert der Rauschspannung U ist abhängig von der Temperatur T , des Widerstandwertes R und der Frequenzbandbreite B der Messung. Es gilt die folgende Gleichung, wobei k_B die Boltzmann-Konstante bezeichnet:

$$U = \sqrt{4k_B \times T \times R \times B} \quad (3.1)$$

Schrotrauschen

Schrotrauschen ist ebenfalls ein weißes Rauschen. Es entsteht, wenn Ladungsträger der Ladung e eine Potentialbarriere überwinden. Der Effektivwert des Rauschstromes ist abhängig von der Frequenzbandbreite B sowie des Eingangsruhestromes I_0 und ergibt sich zu:

$$I = \sqrt{2e \times I_0 \times B} \quad (3.2)$$

1/f Rauschen

1/f Rauschen wird auch ein „Rosa Rauschen“ bezeichnet, und nimmt mit steigender Frequenz ab. Der Frequenzgang verläuft umgekehrt proportional mit 1/f, die Leistungsdichte halbiert sich also bei Verdopplung der Frequenz.

SNR

Eine wichtige Kenngröße beim Umgang mit Signalen und Rauschen ist das SNR (Signal-Rausch-Verhältnis, englisch „Signal to Noise Ratio“). Dieser Begriff aus Elektrotechnik und Messtechnik ist auch eine wichtige Größe bei der Beurteilung der Güte von Lidar-Signalen. Es bedeutet, dass das Verhältnis der Signalleistung S zur Rauschleistung N ist, also $SNR = S/N$. Das SNR ist ein Maß für technische Qualität eines Signals. Es gibt einen Wert in dB an und beschreibt das Verhältnis der Signalspannung U_s zur Rauschspannung U_r :

$$SNR(dB) = 20 \log \frac{U_s}{U_r} \quad (3.3)$$

3. 1.1.3 Bauteilauswahl

Für einen erfolgreichen Aufbau des Messsystems ist es entscheidend, entsprechend der Anforderungen der Anwendung passende Bauteile auszuwählen. Im Nachfolgenden werden Anforderung und daraus resultierende Bauteilauswahl der wichtigen Elemente „Fotodiode“ und „Transimpedanzverstärker“ ausgeführt.

Fotodiode

Abbildung 3.3 informiert sich über die Funktion einer Fotodiode.

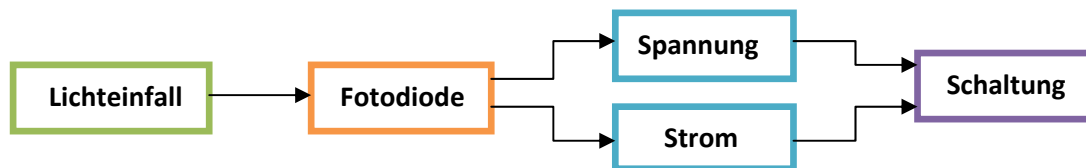


Abbildung 3.3 Funktion der Fotodiode

Eine Fotodiode ist ein Halbleiter-Bauelement. Der p-n-Übergang der Fotodiode ist dem Licht baulich gut zugänglich gemacht. Bei Lichteinfall werden in der Sperrschicht Elektron-Lochpaare erzeugt. Bei Betrieb der Diode in Sperrrichtung bewegen sie sich als freie Ladungsträger aus der Sperrschicht, wodurch der Sperrstrom wächst.

Im Vergleich zu einer normalen p-n Fotodiode ist ein PIN Fotodiode mit einer zusätzlichen i-Schicht (zwischen p und n Schicht) versehen. Die Photonen werden so hauptsächlich in der größeren Raumladungszone absorbiert. Durch die größere Raumladungszone bedingt sich eine kleinere Sperrschichtkapazität. Die Schaltzeit der PIN-Fotodiode verringert sich somit, was den Vorteil kürzerer Ansprechzeiten bietet.

Für den Aufbau einer PIN Fotodiode kommen verschiedenen Halbleitermaterialien, wie z.B. Silizium, Germanium oder InGaAs. Die Photosensitivität einer Silizium-PIN Fotodiode beginnt bei circa 300 nm und endet bei circa 1100 nm. Im Vergleich zu einem Silizium-PIN Fotodiode, haben Germanium- und InGaAs-PIN Fotodioden einen ähnlichen Empfindlichkeitsbereich zwischen 800 nm und 1800 nm haben.

Wichtige Kennwerte einer Fotodiode

1. Empfindlichkeit ist ein Bereich der Wellenlänge, dass die Fotodiode empfangen kann.
2. NEP (englisch: **N**oise **E**quivalent **P**ower) ist ein Maß für die Empfindlichkeit eines optischen Detektor oder Detektor-System. Je geringer NEP die PIN Fotodiode hat, desto besser die PIN Fotodiode ist. Das NEP ist als Verhältnis von Leistung **P** zu SNR messbar.

$$NEP = \frac{P}{SNR} \quad (3.4)$$

3. Die Ansprechzeit ist der Zeitraum von Beginn des Lichteinfalls bis zum Erreichen eines stabilen Ausgangssignals, bis Stabilität der Leistung des Lichteinfalls.

Anforderungen an die auszuwählende Fotodiode

Für die auszuwählende Fotodiode können verschiedene Auswahlkriterien formuliert werden, anhand derer ein passendes Bauelement ausgewählt werden kann. Aus den Anforderungen des Einsatzbereiches des Systems ergeben sich folgende Hauptmerkmale:

1. Möglichst hohe Empfindlichkeit im Bereich von 532 und 1064 nm
2. Geringe NEP
3. Kurze Ansprechzeit

In dieser Arbeit müssen Laserstrahlen in den Wellenlänge 532 nm und 1064 nm detektiert werden, weswegen die Wahl auf eine Silizium Diode fiel. Kurze Ansprechzeiten lassen sich mit PIN-Diode realisieren, wie im vorangegangenen Abschnitt dargestellt wurde. Die Produktrecherche wurde also auf eine Si-PIN Fotodiode eingeschränkt.

Hamamatsu S1223 Si-PIN Fotodiode

Die Produktrecherche führte auf die Fotodiode Hamamatsu S1223, die alle geforderten Eigenschaften aufweist.

1. Spektraler Messbereich: 320 nm- 1100 nm
2. Geringe NEP: $9.14 \times 10^{-15} \text{ W/Hz}^{1/2}$
3. Große Sperrfrequenz: 30 MHz bei 20 V
4. Geringe Dunkelstrom: 0.1 nA (typisch) und 10 nA (maximal) bei 20 V
5. Geringe Terminal-Kondensator: 10 pF bei 20 V 1 MHz

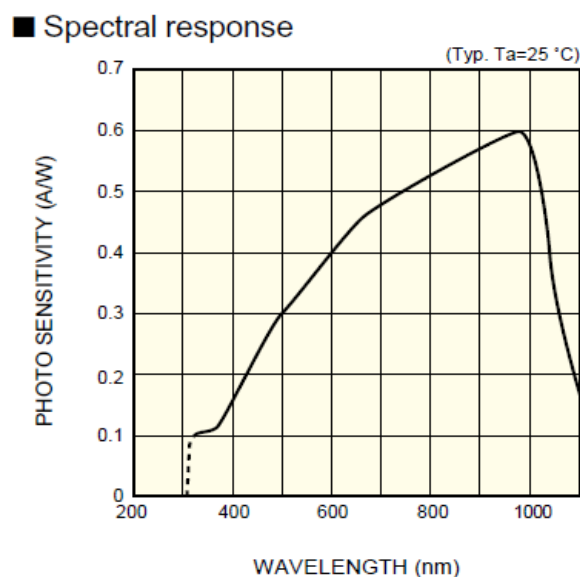


Abbildung 3.4 Spektrale Empfindlichkeit der Hamamatsu S1223 Si PIN Fotodiode
[Datenblatt Hamamatsu S1223]

Die spektrale Empfindlichkeit der Hamamatsu S1223 Si PIN Fotodiode wird in Abbildung 3.4 gezeigt. Dabei ist zu erkennen, dass die höchste Empfindlichkeit mit 0,6 A/W bei etwa 960 nm auftritt. Die Empfindlichkeit bei Wellenlänge 532 nm ist rund 0,35 A/W und bei Wellenlänge 1064 nm ist rund 0,55 A/W (aus der Grafik, typische Betriebstemperatur 25°C).

Die Lichtleistung ist 3 mW bei $\lambda=1064 \text{ nm}$ und 0,04 mW bei $\lambda=532 \text{ nm}$ (bereits gegeben). Es gibt die Gleichung, wobei **S** Empfindlichkeit, **P** Lichtleistung bezeichnet. Nach der Gleichung 3.5 kann der Stromwert der Fotodiode **I** berechnet werden.

$$I = S \times P \quad (3.5)$$

$I(\lambda=1064 \text{ nm}) = 1,65 \text{ mA}$ und $I(\lambda=532 \text{ nm}) = 0,014 \text{ mA}$

Die beiden Ströme sind sehr klein, deswegen ist ein großer Rückkopplungswiderstandswert der Transimpedanzverstärker notwendig, um das schwache Lichtsignal in einem entsprechenden Amplitudenwert umwandeln zu können. (siehe 3.1.1.4)

Transimpedanzverstärker OPA380

Der OPA380 als Transimpedanzverstärker ist ein Hochgeschwindigkeit-Operationsverstärker. Weitere Vorteile seine extrem hohe Präzision, hervorragende Langzeitstabilität. Der OPA380 ist ideal für die Anwendungsgebiete der Fotodiodenüberwachung und des optischen Verstärkers geeignet und wurde daher für die Anwendung ausgewählt.

Eigenschaften:

1. Große Bandbreite: 90 MHz
2. Geringe Offsetspannung: 25 μV maximal
3. Geringe Offsetdrift: 0.1 $\mu\text{V}/^\circ\text{C}$

Die Abbildung 3.5 und 3.6 informieren sich über die schematische Zeichnung des OPA380 und der Transimpedanzverstärker bei $C_{\text{Diode}} = 10 \text{ pF}$.

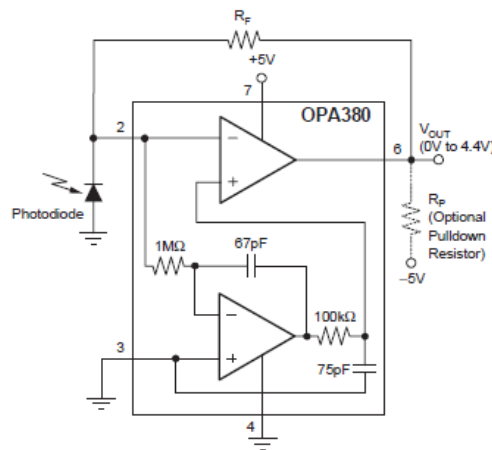


Abbildung 3.5 Schematische Zeichnung der OPA380 [TI OPA380 Datenblatt]

Für den Kapazitätswert von 10 pF der verwendeten Si-PIN Fotodiode Hamamatsu S1223 können der Transimpedanzverstärker des OPA380 aus dem Datenblatt entnommen werden.

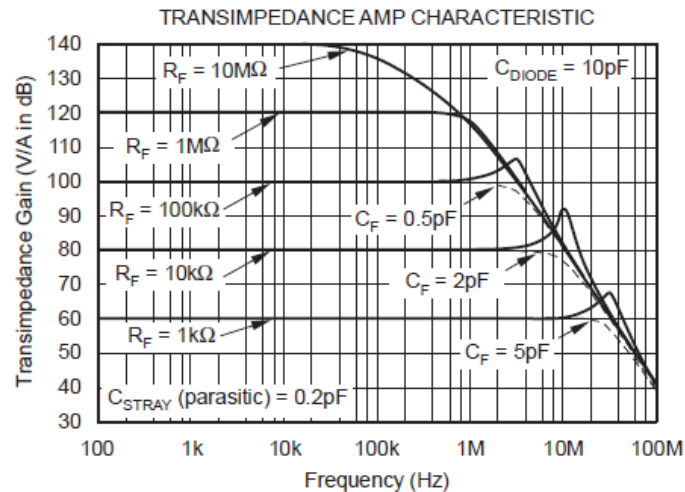


Abbildung 3.6 Transimpedanzverstärker des OPA380 bei $C_{\text{Diode}}=10\text{ pF}$
[TI OPA380 Datenblatt]

3.1.1.4 Schaltung des Transimpedanzverstärkers

Die Beschaltung des Transimpedanzverstärkers wurde auf Grundlage einer der im Datenblatt beschriebenen Grundschaltungen mit den entsprechenden Bauteilen entwickelt.

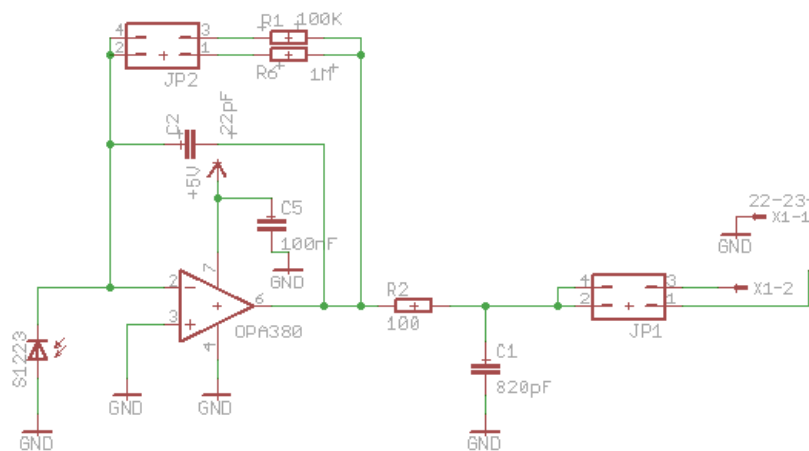


Abbildung 3.7 Schaltung des Transimpedanzverstärkers

- Ganz links ist eine Fotodiode S1223 zu sehen.
- Links Oben befindet sich ein Jumper (JP2), der es möglich zwei unterschiedliche Rückkopplungswiderstandswerte zu wählen: $R1$ (100 k Ω) oder $R6$ (1 M Ω)
- In der mitte stellt der OPA380 und der Stützkondensator $C5$ (100 nF).
- Mit dem Kondensator $C2$ (22 pF) wird die Bandbreite begrenzt.

- Der erste Filter gegen das Hochfrequenz-Rauschsignal wird durch R2 (100 Ω) und C1 (820 pF) realisiert.
- Der Jumper (JP1), auf der rechten Seite, dient zum Ein-oder Ausschalten des Tiefpass-Filters.

3.1.2 Aktive Tiefpass-Filter

3.1.2.1 Funktionsweise

Um nur die nutzbaren Signale zu verarbeiten und die Rauschsignale so klein wie möglich zu halten, wird in der Schaltung ein aktiver Tiefpass-Filter realisiert. Aktive Tiefpass-Filter haben gegenüber passiven Tiefpass-Filter mehrere Vorteile. Sie sind einfacher zu realisieren, da keine Induktivitäten verwendet werden, welche bei tiefen Frequenzen (die Anforderung der Arbeit: Grenzfrequenz ist nur 50Hz) sehr große Werte annehmen würde. Darüber hinaus sind Spulen, relativ zu anderen Bauteilen gesehen, teuer, so dass eine Schaltung bei Verzicht auf Induktivitäten ökonomischer ist.

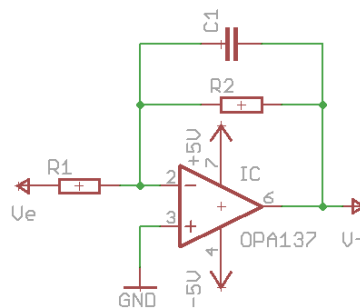


Abbildung 3.8 Grundprinzip eines Tiefpass-Filters

Die Abbildung 3.8 zeigt, wie eine aktive Tiefpass-Filterschaltung mit einem Operationsverstärker und einem RC Beschaltungsnetzwerk prinzipiell aufgebaut werden kann.

Die Ordnung des aktiven Tiefpass-Filters spielt eine große Rolle. Je höher die Ordnung ist, desto steiler ist der frequenzabhängige Abfall des Signals. Für eine praktische zu realisierende Anwendung werden häufig Filter der 2. oder 3.Ordnung eingesetzt. Ein Tiefpass 2.Ordnung reicht wegen der Anforderung dieser Arbeit aus[siehe 3.1.2.3].

Die Übertragungsfunktion $A(s)$ eines Tiefpasses der 2. Ordnung kann über die Relation berechnet werde. Dabei ist ω_g die Grenzfrequenz der Übertragung, a_1 und b_1 sind reelle Koeffizient und s ist komplexe Frequenz.

$$A(s) = \frac{1}{(1+a_1s+b_1s^2)} \quad \text{mit} \quad S = \frac{s}{\omega_g} \quad (3.6)$$

Es gibt hauptsächlich drei verschiedene Charakteristiken eines aktiven Tiefpass-Filters, je nach Wahl der Koeffizienten a_1 und b_1 . Sie sind Butterworth-Filter, Bessel-Filter und Tschebyscheff-Filter.

1. Das Butterworth-Filter sorgt für einen maximalen Durchlassbereich.
2. Der Bessel-Filter hat unterhalb der Grenzfrequenz ein optimales Rechteckübertragungsverhalten[Heinz Schmidt-Walter].
3. Der Tschebyscheff-Filter. Der Amplitudengang hat im Durchlassbereich eine definierte Welligkeit. Dadurch wird der Abfall oberhalb der Grenzfrequenz besonders steil[Heinz Schmidt-Walter].

Die Koeffizienten und Gütefaktor (Q Faktor: Maß für die Dämpfung eines schwingfähig System) hängen vom Typ des Tiefpass-Filters ab. Die Tabelle 3.1 zeigt die Koeffizienten und den Gütefaktor mit drei verschiedenen Tiefpass-Filtern.

Typ	a_1	b_1	Gütefaktor Q
Butterworth	1,4142	1,0000	0,7100
Bessel	1,3617	0,6180	0,5800
Tschebyscheff	1,3614	1,3827	0,8600

Tabelle 3.1 Übersicht drei Type der Tiefpass-Filtern

3.1.2.2 Bauteilauswahl

Operationsverstärker OPA137 für den Tiefpass-Filter

Der OPA137 wird von Texas Instruments hergestellt und oft im Bereiche der aktiven Filter oder Fotodetektor-Verstärker verwendet. Er zeichnet sich durch eine große Bandbreite, schnelle Anstiegsrate und geringes Rauschen aus. Außerdem besitzt er auch einen größeren Versorgungsspannungsbereich im Vergleich zu anderen Produkten, wie z.B. OPA365.

Eigenschaften:

1. Geringe Offsetspannung: 1,5 mV
2. Geringe Eingangsstrom: 1 pA
3. Versorgungsspannungsbereich: $\pm 2,25$ V bis ± 18 V

3.1.2.3 Simulation des Schaltungsverhaltens

Wie vorher das Prinzip eines aktiven Tiefpass-Filters erklärt wurde, eine passende Art eines Tiefpass-Filters zu wählen, die alle anderen Komponentenwerte berechnen, ist nicht einfach. Mit einer speziellen simulierten Software FilterProTM von Texas Instruments braucht man nur 4 Schritten, um einen aktive Tiefpass-Filter schneller und einfacher aufzubauen.

1. Aktiven Pass-Filter-Typ auswählen (hier: Tiefpass).

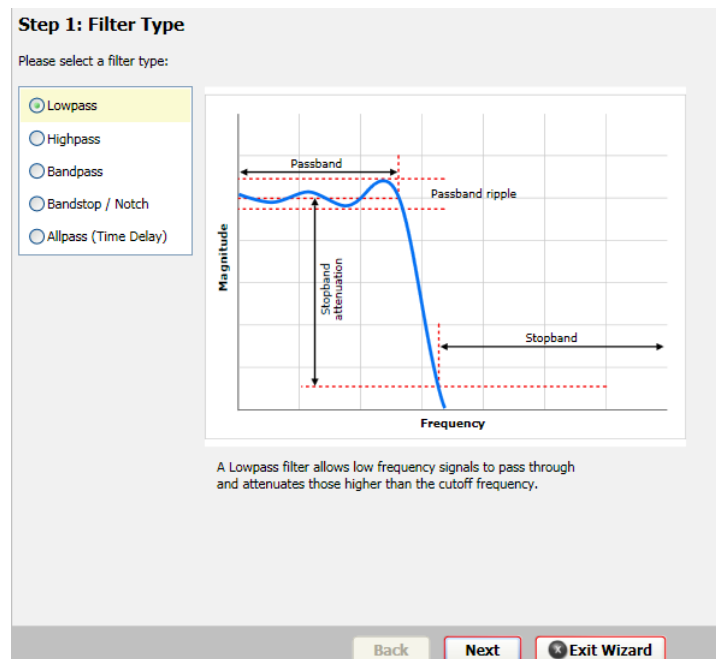


Abbildung 3.9a Simulationssoftware 1 Schritt

2. die wichtige Informationen abgeben (hier: Grenzfrequenz: $f=50$ Hz, Anforderung meiner Arbeit feststellen).

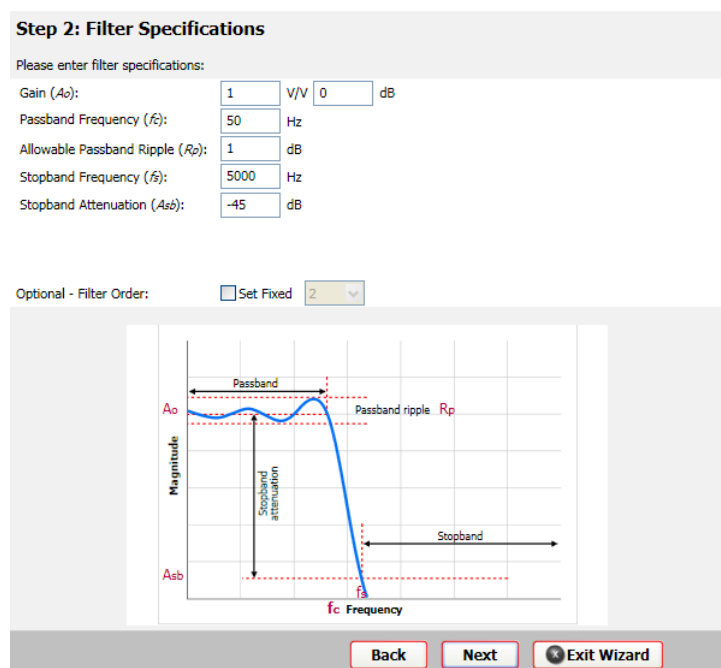


Abbildung 3.9b Simulationssoftware 2 Schritt

3. Aktiven Tiefpass-Filter-Response-Typ (hier: Bessel 2.Ordnung) auswählen. Diese Simulationssoftware bietet die beste Art bzw. die beste Ordnung eines aktiven Tiefpass-Filters an. Deswegen wurde hier Bessel 2.Ordnung ausgewählt.

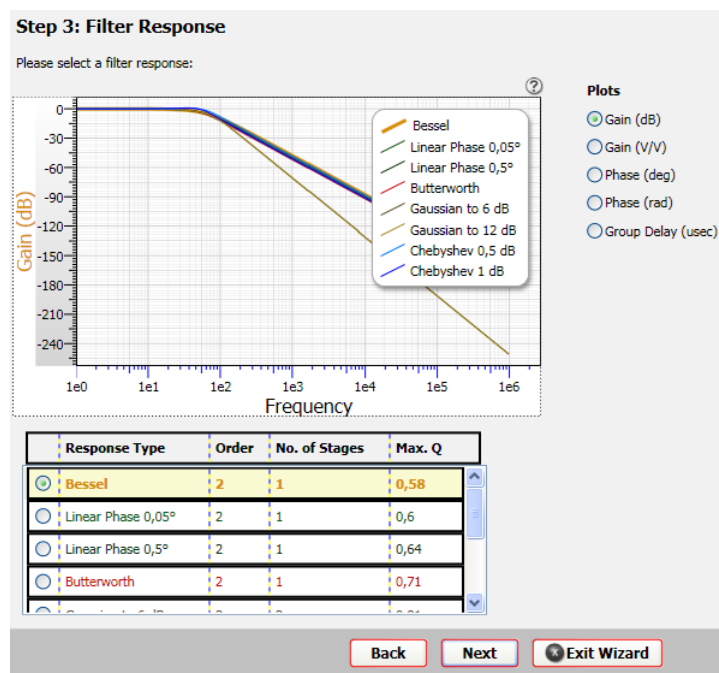


Abbildung 3.9c Simulationssoftware 3 Schritt

4. Aktive Tiefpass-Filter-Topologie (hier: Multiple-Feedback) auswählen.

Es gibt zwei aktive Tiefpass-Filter-Topologien: Multiple-Feedback Topologie und Sallen-Key Topologie. Obwohl die Sallen-Key Topologie weniger Bauteile braucht, z.B. bei dieser Schaltung nur zwei Widerstände, hat es auch zwei Nachteile. Erstens: Es hat eine hohe Empfindlichkeit gegenüber Toleranzen der Bauteile. Zweitens: Bei der Messung betreffen sowohl die Filtercharakteristik als auch die Verzerrungen. Daher wurde die Multiple-Feedback Topologie in dieser Schaltung gewählt.

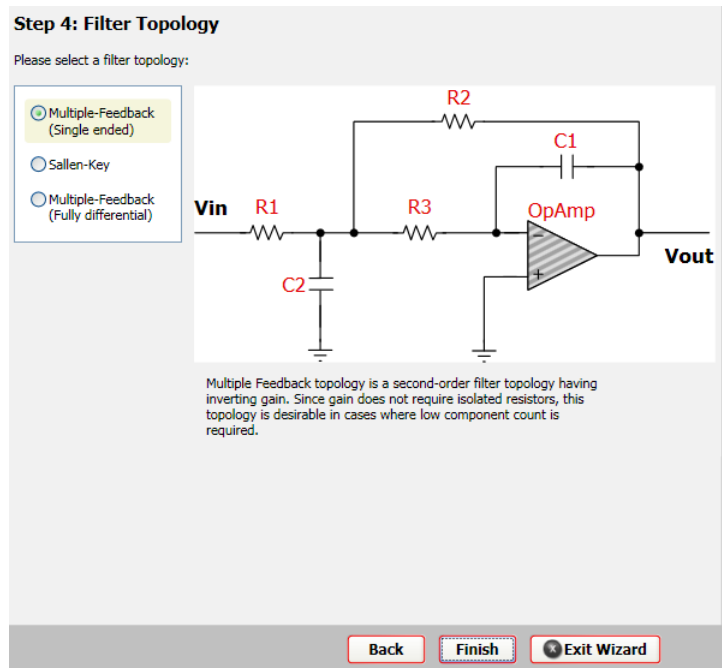


Abbildung 3.9d Simulationssoftware 4 Schritt

Wenn man zum Schluss „Finish“ klickt, bekommt man das Ergebnis „Tiefpass mit Bessel 2.Ordnung, Multiple-Feedback Topologie“ und die Grenzfrequenz ist 50 Hz mit -3dB gedämpft.

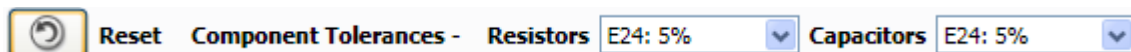


Abbildung 3.9e Toleranzauswahl der passiven Bauelemente

Weil die Widerstände und die Kondensatoren E24 (Toleranz 5%) sind, wurde durch die 4 Schritte der Software die mögliche Schaltung und die Werte der passiven Bauelemente gegeben, wie Abbildung 3.9f und 3.10 zeigt:

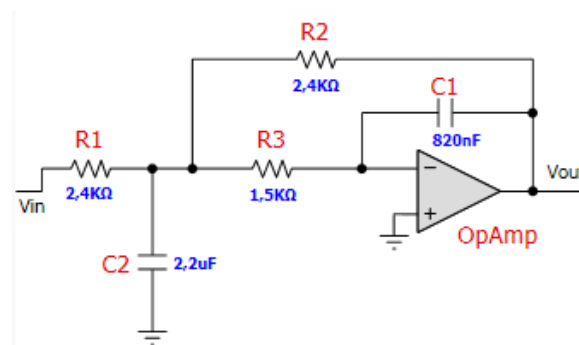


Abbildung 3.9f Vorgeschlagene Schaltung

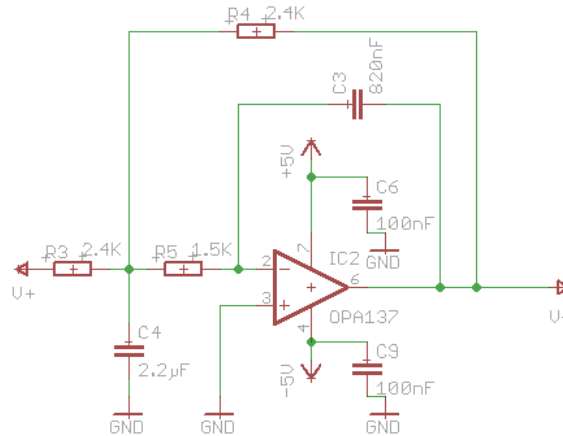


Abbildung 3.10 Schaltung des aktiven Tiefpass-Filters

- Ganz links ist die Eingangsspannung vom Fotodiodensignalverstärker.
- Die anderen passiven Bauelemente (Widerstände und Kondensatoren), wie Software simuliert, bekommen eine aktive Tiefpass-Filter-Schaltung (Bessel 2.Ordnung).
- In der Mitte befindet sich der OPA137 und die Stützkondensatoren C6 (100 nF) und C9 (100 nF), um eine positive und eine negative Spannungsversorgung zu erhalten. Die Verstärkung ist hier -1 ($\text{Gain} = -\frac{R4}{R3} = -1$).
- Ganz rechts wird deshalb eine negative Ausgangsspannung zum Anschluss an das ADAM4017 Modul erzeugt.

Der Frequenzgang des simulierten Tiefpass-Filters mit diesen Werten ist in den unteren Abbildungen dargestellt.

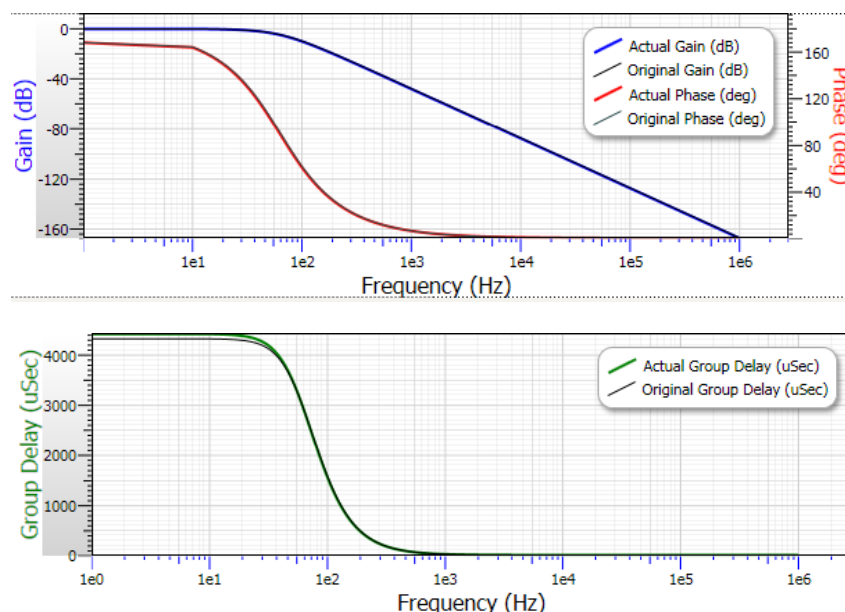


Abbildung 3.11 Frequenzgang des simulierten Tiefpass-Filters

3.1.3 Spannungsversorgung

Natürlich muss die Schaltung versorgt werden. In dieser Schaltung müssen zwei unterschiedliche Spannungen zur Verfügung stehen, sowohl +5 VDC als auch -5 VDC. Dafür nutze ich ein DC-DC Wandler, der mir die gewünschten Spannungen (von +12 VDC auf ± 5 VDC) zur Verfügung stellt.

3.1.3.1 Funktionsweise

Um eine gewünschte konstante Spannung zu erhalten, wird der DC-DC Wandler verwendet. Der DC-DC Wandler erzeugt aus Gleichspannung eines bestimmten Bereichs eine konstante Ausgangsspannung mit nur geringer Welligkeit, die zur Versorgung von der Schaltung benutzt werden kann. Die Eingangsspannung in der Arbeit ist +12 VDC. Weil eine Spannung von ± 5 VDC nötig ist, wird ein Gleichspannungswandler verwendet, mit dem ein niedrigeres und invertiertes Spannungsniveau erzeugt werden kann.

3.1.3.2 Bauteilauswahl

Traco Power TEL-2-1221

Der Traco Power TEL-2-1221 ist ein isolierter 2W DC-DC Wandler. Bei nur sehr kleinem Oberflächenbedarf bietet er ohne zusätzliche Komponenten einen kompletten DC-DC Wandler. Traco Power TEL-2-1221 ist durch den weiten Eingangsbereich und die genau regulierten Ausgänge für meine Arbeit interessant. Außerdem ist er auch preiswert.

Eigenschaften:

1. Eingangsspannung: von 9 VDC bis 18 VDC
2. Ausgangsspannung: ± 5 VDC
3. Ausgangsstrom: ± 200 mA
4. E/A-Isolation: 1500 VDC
5. Niedrige Restwelligkeit

3.1.3.3 Schaltung der Spannungsversorgung

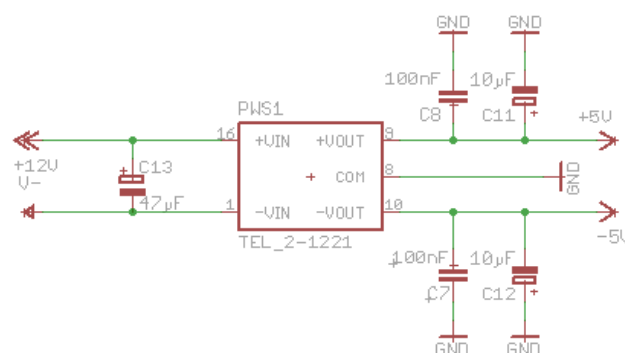


Abbildung 3.12 Schaltung der Spannungsversorgung

- Auf der linken Seite gibt es eine Eingangsspannung von +12 VDC.
- C13(Tantal-Kondensator 47 μ F) als Glättungskondensator befindet sich vor dem DC-DC Wandler. Der DC-DC Wandler benötigt im Leistungsteil im Allgemeinen Kondensatoren, welche in Form von Tantal- oder Elektrolytkondensatoren ausgeführt sind. Er sollte so nah wie möglich vor dem DC-DC Wandler stehen und seine Funktion ist Minderung der Restwelligkeit.
- In der Mitte erkennt man den DC-DC Wandler Traco Power TEL-2-1221.
- zwei Keramik-Kondensatoren (C7 und C8 100 nF) als Stützkondensatoren befinden sich hinter dem DC-DC Wandler.
- zwei Tantal-Kondensatoren (C11 und C12 10 μ F) als Glättungskondensatoren befinden sich stellen durch Parallelschaltung hinter den beiden Keramik-Kondensatoren. Die C11-Anode ist am positiven Pol der Speisespannung angeschlossen und die C12-Anode ist an die Masse angeschlossen. Die beiden Werte Tantal-Kondensatoren sind 10 μ F (üblicher Wert). Sie verhindern die Schwingneigung.
- Am Ausgang bekommt man zwei Ausgangsspannungen +5 VDC (Pin9) und -5 VDC (Pin8).

3.1.4 Gesamtschaltplan

Die Si-PIN Fotodiode Hamamatsu S1223 dienen zur Aufnahme dieser Signalquelle. Sie liefern ein sehr schwaches, analoges Signal. Das Signal wird als erstes durch den OPA380 verstärkt. Weil es problematische Rauschsignale gibt, wird ein Filter hinzugefügt. Danach wird dieses Signal durch einen optionalen aktiven Tiefpass-Filter gefiltert. Nun sollten noch Frequenzen bis 50 Hz vorhanden sein. Das Signal wird in den A/D Wandler (ADAM 4017 Modul) eingespeist. Das digitale Signal des ADAM4017 kann man durch Delphi 6 Programmierung auf dem Computer für den Zweck benutzen.

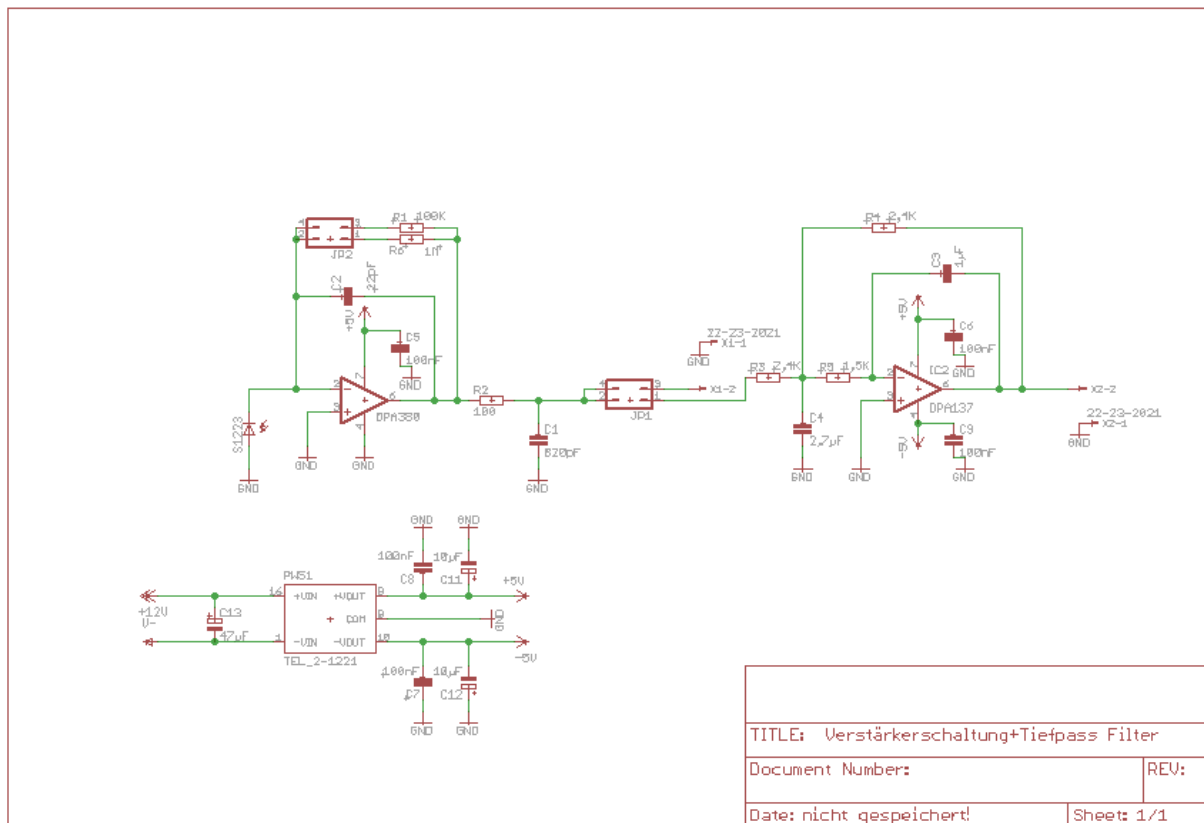


Abbildung 3.13 Gesamtschaltplan für den Fotodiodenverstärker

3.1.5 Fertigung dieser Platine

Nach der Entwicklung dieser Schaltung mit dem EDA-Programm Eagle, habe ich die Platine mit Mikroskop gelötet. In meiner Arbeit wurden die SMD-Bauelemente bevorzugt, weil sie weniger Platz einnehmen und kostengünstiger sind.

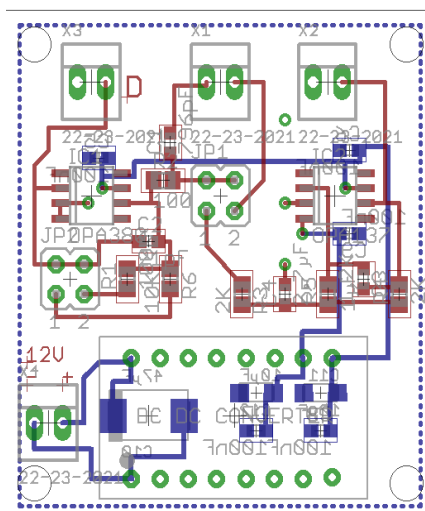


Abbildung 3.14 PCB Layout und gelötete Platine für den Fotodiodenverstärker

In der oberen Ebene (rot) stellt die Signalleistung und in der unteren Ebene (blau) stellt die Spannungsversorgung. Ausnahme ist DC-DC Wandler. Die Bauelemente, die in dieser Schaltung benutzt wurden, sind in der Tabelle 3.2 aufgeführt.

Bauelemente	Gehäuse-Typ	Anzahl
OPA380	SMD SOIC8	1
OPA137	SMD SOIC8	1
Keramik-Kondensator	SMD 0603	1(820 pF;22 pF;1 μ F;2,2 μ F);5(100 nF)
Tantal-Kondensator	SMD SMC A	2(10 μ F)
Tantal-Kondensator	SMD SMC D	1(47 μ F)
Jumper	JP2Q	2
Widerstand	SMD 0805	1(100 Ω ;1,5 k Ω ;100 k Ω ;1 M Ω);2(2,4 k Ω)
DC-DC Wandler	Traco Power 2W	1

Tabelle 3.2 alle Bauelemente der Platine für den Fotodiodenverstärker

3.2 Platine für die Temperaturmessung

3.2.1 Funktionsweise

Die anderen Parameter für Laserüberwachung sind Temperaturen. Man muss auch eine Parallelschaltung aufbauen, um sich über die vier Temperaturparameter zu informieren. Der Sinn dieser Schaltung liegt darin, Spannungsänderungen (abhängig von Temperaturänderungen) durch Auslesen des ADAM Moduls auf dem Rechner sichtbar zu machen. Bei der Schaltung mit dem Heißeiter handelt sich um einen Widerstand, der linear zur Temperatur seinen Widerstandswert verändert. Nach einer bestimmten Gleichung kann man die Temperatur berechnen. Um durch eine Widerstandsänderung eine Spannungsänderung hervorzurufen, wird eine Art Spannungsteiler verwendet, d.h. wenn man die Spannungsänderungen gemessen hat, weiß man die Temperaturänderungen.

3.2.2 Bauteilauswahl

Heißeiter

Bei der Temperaturmessung werden Heißeiter sehr oft als Temperatursensor benutzt. Heißeiter sind keramische passive Bauelemente, deren elektrischer Widerstand bei steigender Temperatur nichtlinear abnimmt. Deswegen nennt man Heißeiter auch NTC-Widerstand (Negative Temperature Coefficient-Widerstand). Aufgrund ihrer hohen Empfindlichkeit und Zuverlässigkeit sind Heißeiter für den Einsatz als Temperatursensoren prädestiniert. Weil bei steigender Temperatur mehr Elektronen aus ihren Kristallbindungen herausgerissen werden, leiten Heißeiter bei hoher Temperatur besser als bei niedrigen Temperaturen.

In der Arbeit wurden vier NTC-Widerstände vom Hersteller EPCOS AG (Typ:B57861) eingesetzt. Der Grund ist, dass sie kurze Ansprechzeiten und hohe Genauigkeiten haben.

Eigenschaften:

1. mit Epoxidharz umhüllt und versilberte Ni-Drähte
2. Widerstandswert ist 100 kΩ bei einer Temperatur von 25°C
3. Temperaturmessbereich: -55°C bis 155°C

Der temperaturabhängige Widerstand dieses Heißleiters lässt sich beschreiben mit der folgenden Gleichung:

$$T = \frac{B \cdot T_{25}}{B + \ln\left(\frac{R}{R_{25}}\right) \cdot T_{25}} \quad (3.7)$$

Die einzelnen Formelzeichen stehen für folgende Größe:

- T: Betriebstemperatur
- B: Beta-Wert/Materialkonstante (hier:4540 im Datenblatt angegeben)
- T₂₅:Nenntemperatur (hier:25°C)
- R: Widerstandswert bei der Betriebstemperatur
- R₂₅:Nennwiderstandswert bei der Nenntemperatur

Wenn Werte von B, T₂₅, R, R₂₅ vorhanden sind, kann man durch diese Gleichung die Betriebstemperatur berechnen.

AD586

Der Operationsverstärker AD586 von ANALOG DEVICES wurde in der Halbbrücke-Schaltung der Temperaturmessung verwendet. Die Versorgung der Dehnungsmessstreif-Brücke wird von der Konstantspannungsquelle AD586 geliefert. AD586 hat eine Skalierung von 10,2 mV pro °C. Mit NTC100 kann man durch die Spannungsänderung die Temperatur berechnen.

Eigenschaften:

1. Versorgungsspannung: 5 V
2. Geringes Rauschen: 100 nV√Hz
3. Geringe Offsetspannung: 42 mV
4. Messbarer Temperaturbereich: -55°C bis 125°C

3.2.3 Schaltung der Temperaturmessung

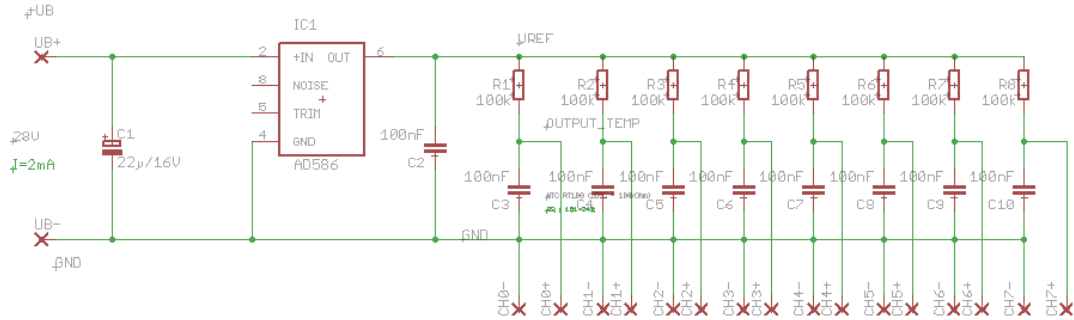


Abbildung 3.15 Schaltung der Temperaturmessung

Diese Schaltung wurde von Herrn Köpnick (Ingenieur am IAP) entwickelt und von mir gelötet und gefertigt.

- Auf der linken Seite gibt es eine originale Eingangsspannung +12 VDC.
- C1 (Tantal-Kondensator 22 µF) und C2 (Keramik-Kondensator 100 nF) als Stützkondensatoren stellen vor bzw. nach dem AD586. Ihre Funktion ist die Verminderung der Eingangsspannung von +12 VDC auf +5 VDC. $U_{ref}=+5$ VDC.
- In der Mitte befindet sich der Operationsverstärker AD586.
- besteht aus 8 gleichen Widerständen (je 100 kΩ) und 8 gleichen Kondensatoren (je 100 nF) die Parallelschaltung, damit jede Pin eine gleiche Spannung von +5 VDC bekommen kann.
- Die Funktionen von CH0- bis CH7+ sind Anschluss an das ADAM4017 Modul, um Temperaturen zu messen (hier werden nur CH1- bis CH3- benutzt).

Durch diese Schaltung ergeben sich die Gleichungen für die Temperaturmessung:

$$U_o = U_{ref} - U_{NTC} \quad (3.8)$$

$$I_o = U_o / R_o \quad (3.9)$$

$$I_a = U / R_a \quad (3.10)$$

$$I_{NTC} = I_o - I_a \quad (3.11)$$

$$R_{NTC} = U_{NTC} / I_{NTC} \quad (3.12)$$

Die einzelnen Formelzeichen stehen für folgende Größe:

- U_{ref} : Eingangsspannung (hier 5 V)
- U_o : Spannung an dem Parallelwiderständen (hier R1 bis R8)

- U_{NTC} : NTC-Spannung
- R_{NTC} : NTC-Widerstandswert
- R_a : Innerwiderstand des ADAM4017 Moduls (hier 5 M Ω)
- R_o : Widerstandswert (hier R1 bis R8, 100 k Ω)
- I_{NTC} : NTC-Strom
- I_o : Strom an den Parallelwiderständen (hier R1 bis R8)
- I_a : Strom am Innerwiderstand des ADAM4017 Moduls

3.2.4 Fertigung dieser Platine

Nach der Entwicklung dieser Schaltung mit dem EDA-Programm Eagle, wurde diese Platine mit Mikroskop gelötet. In meiner Arbeit wurden SMD-Bauelemente bevorzugt, weil sie weniger Platz einnehmen und kostengünstiger sind.

Alle Bauelemente stehen auf einer Seite, wie Abbildung 3.16 zeigt.

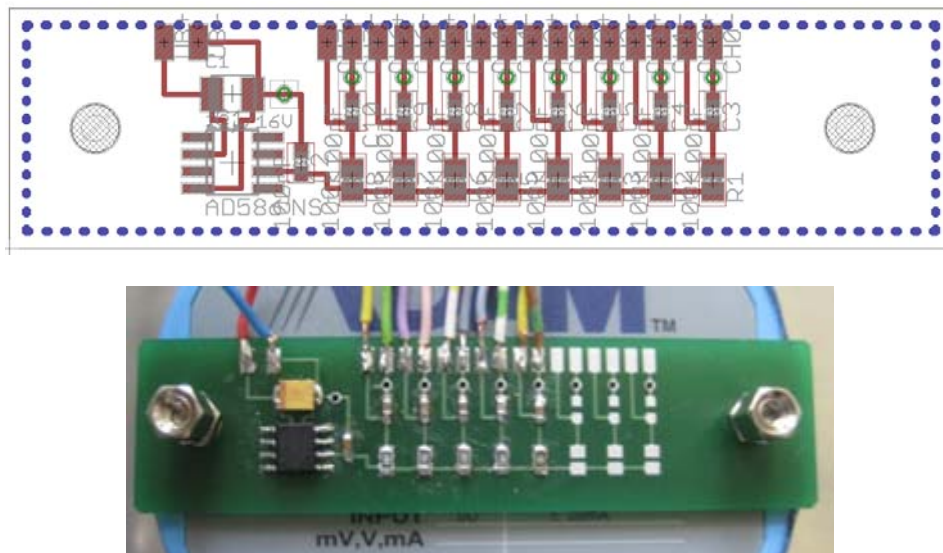


Abbildung 3.16 PCB Layout und gelötete Platine für die Temperaturmessung

Weil es nur wenige Bauelemente gibt, ist eine Ebene genug. Die Bauelemente, die in dieser Schaltung benutzt wurden, sind in der Tabelle 3.3 aufgeführt.

Bauelemente	Gehäuse-Typ	Anzahl
AD586	SMD SO08	1
Keramik-Kondensator	SMD 0603	9(100 nF)
Tantal-Kondensator	SMD SMC 8	1(22 μ F)
Widerstand	SMD 0805	8(100 k Ω)

Tabelle 3.3 alle Bauelemente der Platine für die Temperaturmessung

3.3 Aufbau der Gehäuse und Netzteil

3.3.1 Gehäuse

Ein geeignetes Gehäuse ist sehr wichtig, um das Laserüberwachungsinstrument auf einen passenden Platz im Labor stellen zu können. Gleichzeitig sollte es ein gutes Gehäuse und andere Funktionen bzw. Vorteile haben, z B geringes Gewicht, Staubgeschützt, Spritzwassergeschützt usw. Nach einigem Suchen habe ich ein Aluminium-Gehäuse von Shroff gefunden. Mit dem Software Frontplatten Designer 4.1 der Schaeffer AG kann dieses Gehäuse einfach entwickelt werden. Nach eigenem Design kann es online bestellt werden. In der Arbeit ist das Gehäuse 44 cm lang, 28 cm breit und 8,8 cm hoch.

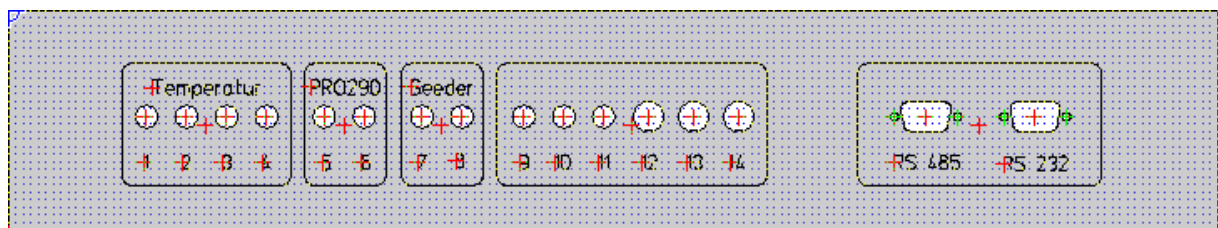


Abbildung 3.17 Gehäuse-Vorderseite

Die Abbildung 3.9 zeigt die Frontplatt-Laserüberwachung. Die verschiedenen Zahlen haben folgende Bedeutung:

Temperatur

1. Laser-Kopftemperatur
2. Raumtemperatur
3. Kühlwassertemperatur
4. frei für zukünftige Nutzung

PRO290

5. BUTR(Build-up-time-reduction)-Spannung
6. Piezo-Spannung

Seeder

7. Fotodioden-Spannung 1
8. Fotodioden-Spannung 2

RS 232: Kommunikation zum Rechner

von Nr.9 bis Nr.14 sowie RS485 sind momentan nicht belegt.

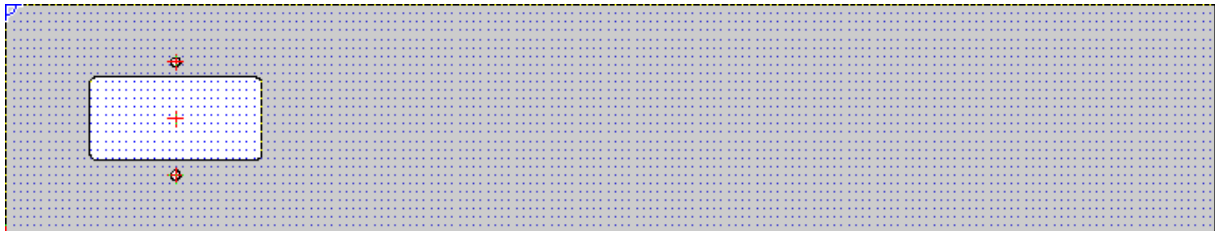


Abbildung 3.18 Gehäuse-Rückseite

Abbildung 3.18 zeigt Gehäuse-Rückseite. Es gibt einen Platz, um das Netzteil anzuschließen.

Nach mechanischem Aufbau wurde das Gehäuse in der Arbeit eingesetzt.

3.3.2 Netzteil

Das Netzteil muss das Messsystem mit der Spannung versorgen. Es besteht aus einem Einbausteckverbinder und einem AC-DC Wandler. Der Einbausteckverbinder ist vom Typ Schaffner FN 393 und wie die Sicherung schon vorhanden. Für die ADAM Module und für die beiden Platinen wird eine Spannung von 12 VDC benötigt. Es wird ein AC-DC Wandler von Traco-Power verwendet, der 220 VAC in 12 VDC umwandelt.

3.4 ADAM Module

3.4.1 ADAM4017 Modul

Ein ADAM Modul wurde vorher in der Laser-Messtechnik am Leibniz-Institut für Atmosphärenphysik e.V. für das mobile Eisen-Lidar erfolgreich eingesetzt. Allein deshalb ist es sinnvoll, ein ADAM4017 Modul für die Laserüberwachung des neuen RMR-Lidars einzusetzen.

Ein ADAM4017 Modul ist ein Kommunikations-und I/O Modul. Es hat 8 analoge Eingänge für mV, V und A, die galvanisch getrennt sind.

Eigenschaften:

1. 8 galvanisch getrennte analoge Eingänge mit 8 16-Bit Auflösung
2. 6 differential +2 single ended Kanäle
3. 3000 V Isolationsschutz und Überspannungsschutz
4. Watchdog Timer
5. RS-485 Schnittstelle und bis zu 256 Module im RS-485 Netz adressierbar
6. Spannungsversorgung: 10 V bis 30 V
7. Messbereich programmierbar: ± 150 mV, ± 500 mV, ± 1 V, ± 5 V, ± 10 V und ± 20 mA
8. Abtastrate: 10 Messungen/Sek
9. Bandbreite: 13,1 Hz
10. Genauigkeit: $\pm 0,1\%$ oder besser

Die Beschreibung der Kanäle des ADAM4017 Moduls zeigt folgende Tabelle:

Kanal-Nr.	Beschreibung
1	Laser-kopf-Temperatur
2	Raum-Temperatur
3	Kühlwasser-Temperatur
4	frei
5	BUTR-Spannung
6	Piezo-Spannung
7	Fotodioden-Spannung 1
8	Fotodioden-Spannung 2

Tabelle 3.4 Beschreibung der Kanäle des ADAM4017 Moduls

3.4.2 ADAM 4520I Modul

Ein ADAM4520I Modul ist ein Konverter Modul. Mit ihm kann man den ADAM4017 mit dem Computer verbinden.

Eigenschaften:

1. Eingang: RS-232 Schnittstelle (4-Wire)
2. Ausgang: RS-485 Schnittstelle (2-Wire) und RS-422 Schnittstelle (4-Wire)
3. 3000V Isolationsschutz und Überspannungsschutz
4. Spannungsversorgung: 10 V bis 30 V

Das ADAM4017 Modul kann als A/D Wandler arbeiten. Die Datenprogrammierung erfolgt in der Sprache Delphi 6 (Borland). Nach der Kommunikation und Programmierung mit dem Computer wurden die Daten angezeigt und weiter verarbeitet. Auf die Funktionsweise wird in Kapitel 4 eingegangen.

3.5 Gesamtplan des Messsystems

Abbildung 3.19 zeigt den Gesamtplan der Laserparameterüberwachung des RMR-Lidars am IAP, die Tabelle 3.5 zeigt alle physikalischen Bauteile und Abbildung 3.20 zeigt die Gehäuse des Messsystems.

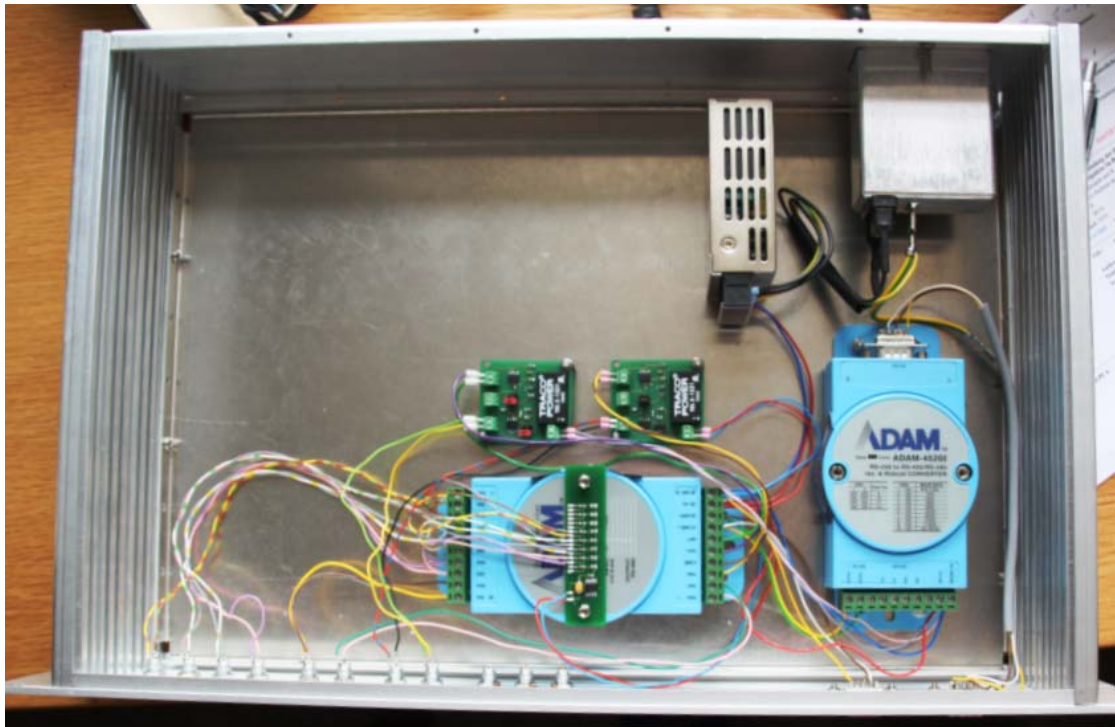


Abbildung 3.20 Gehäuse: Draufsicht der aufgebauten Hardware

3.6 Hardware Test

Um sicherzustellen, dass die Platinen den Anforderungen genügen, müssen nach dem Löten die zwei entwickelten Leiterplatten getestet werden.

3.6.1 Platine für den Fotodiodenverstärker

Der Test der Platine für den Fotodiodenverstärker besteht aus zwei Schritten: den Test der Spannungsversorgung und den Test der Signalleistung.

Spannungsversorgung

Zuerst wurde der Teil der Spannungsversorgung getestet. Wenn eine konstante Eingangsspannung von +12 V vom Netzgerät abgegeben wird, sollen die Pins 9 des DC-DC Wandlers eine Ausgangsspannung von +5 V und die Pins 10 des DC-DC Wandlers eine Ausgangsspannung von -5 V bekommen. Die Werte wurden mit dem Multimeter richtig angezeigt. Danach müssen auch die Eingangsspannungen der beiden Operationsverstärker gemessen werden, +5 V bei den Pins 7 von OPA380 und OPA137, -5 V bei den Pins 4 von OPA137. Wenn alle Spannungen richtig sind, wird der Test der Signalleistung gestartet.

Signalleistung

Danach wurde der Teil der Signalleistung getestet. Mit dem Seederlaser (Wellenlänge $\lambda=532$ nm) kann die Wellenform der Platine am Oszilloskop abgelesen werden. Durch das Signal aus

dem Seederlaser, das von stark bis schwach variiert, wurde die Änderung der Wellenform am Oszilloskop beobachtet, d.h. die Ausgangsspannung wurde auch immer kleiner. Um ein schwaches Signal besser anzeigen zu können, muss eine größere Verstärkung zur Verfügung gestellt werden. Nach dem Vergleich zwischen beiden Widerstandswerten ($1\text{ M}\Omega$ und $100\text{ k}\Omega$) wurde der Widerstand von $1\text{ M}\Omega$ aufgrund der besseren Wellenform eingesetzt. Der nächste Schritt ist der Test des Tiefpass-Filters. Wenn das System mit dem Tiefpass-Filter läuft, kann die Frequenz über 50 Hz gefiltert werden und gleichzeitig erhält man eine negative Ausgangsspannung zwischen -5 V und 0 V . Das ist ein besseres, saubereres Signal.

3.6.2 Platine für den Temperaturmessung

Im Vergleich zum Test der Platine für Fotodiodenverstärker ist der Test der Platine für Temperaturmessung einfacher. Für diese Parallelschaltung misst man nur die vier eingesetzten Kanäle (von CH1 bis CH4) und vergleicht, ob alle eine gleiche Spannung von $+5\text{ V}$ haben. Bei der gleichen Spannung am Anschluss zum ADAM Modul und mit vier NTC 100 (Heißleiter) kann man später richtig die Temperaturen mit einem Programm berechnen.

Die beiden Platinen wurden nach dem Test an den ADAM Module angeschlossen.

4. Software-Design

Nach dem Hardware Entwurf wurde das Software-Design entwickelt. In dem folgenden Kapitel wird der Quellcode anhand von Flussdiagrammen beschrieben. Der Quellcode selber befindet sich am Ende in Anhang der Arbeit.

4.1 Idee und Programmablaufplan

Die Software Delphi 6 wird am IAP bereits benutzt und ist als im Projekt einzusetzendes Produkt in der Masterarbeit festgelegt. Delphi 6 ist auf PASCAL aufgebaut. Obwohl Delphi 6 eine alte Software (seit 2001) ist und vorher unbekannt für mich war, habe ich mit Delphi 6 programmiert, um die Kompatibilität zum vorhandenen System zu haben.

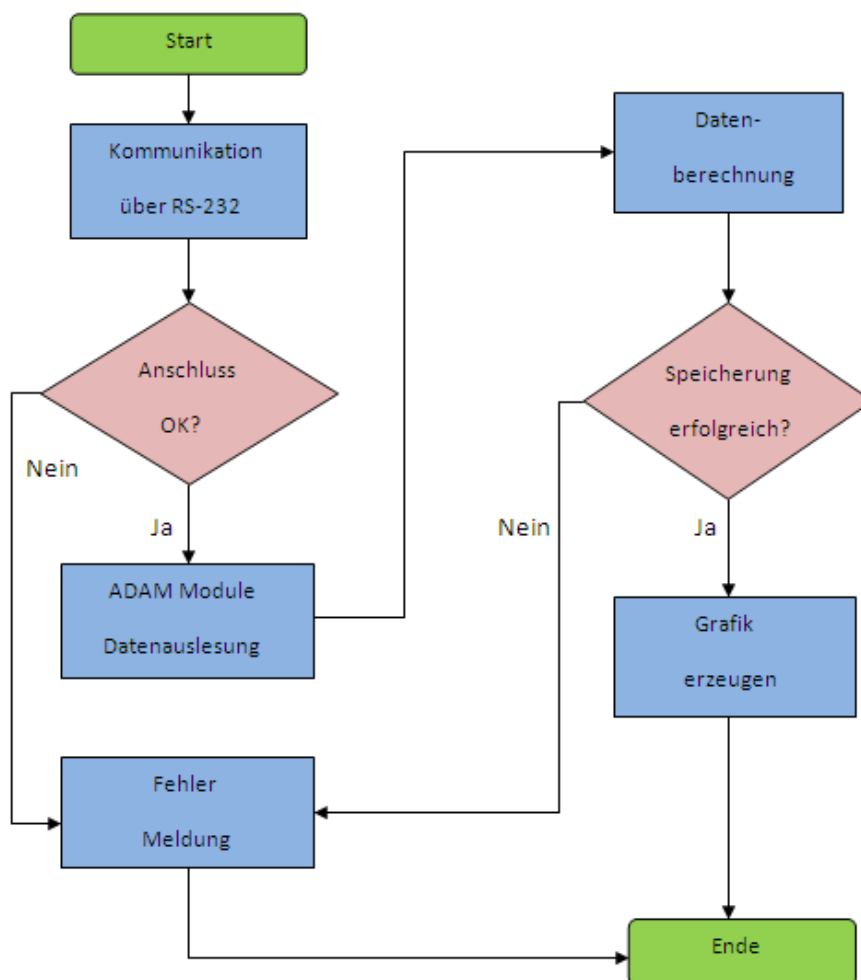


Abbildung 4.1 Programmablaufplan mit Delphi 6

Das obere Flussdiagramm gibt den Programmablauf an. Es besteht hauptsächlich aus fünf Teilen: Datenübertragung, Datenauslesen, Datenberechnung, Datenspeicherung sowie der

Grafikerzeugung. Eine Erklärung der Einzelschritte sowie die jeweilige Benutzeroberfläche erfolgt im nächsten Schritt.

4.2 Benutzeroberflächen

4.2.1 Datenübertragung

Die ADAM Module werden über den COM-Port des Computers gesteuert. Um die Kommunikation zwischen Computer und ADAM Module herzustellen, wird eine RS-232 Schnittstelle eingesetzt. Zur Ansteuerung der ADAM Module existiert bereits eine Software „Advantech Adam.Net Utility“. Die Software ermöglicht einen Datenzugriff: Baudrate 9600, 8 Databits, 1 Stopbits und Timeout 1000 ms. Die bereits vorhandene Delphi-Komponente „myCom“ wird verwendet. Die Tabelle 4.1 zeigt die Pins und die Funktion einer RS-232 (9-poligen) Schnittstelle.

Pin-Nr.	Name	Beschreibung
1	Data Carrier Detect (DCD)	Signalträger erkannt
2	Receive Data(RxD)	Empfangsdaten
3	Transimit Data (TxD)	Sendedaten
4	Data Terminal Ready(DTR)	Daten Terminal bereit
5	Ground (GND)	Signalmasse
6	Data Set Ready (DSR)	Daten Endgerät bereit
7	Request to Send (DTS)	Sendeanfrage
8	Clear to Send (CTS)	Senden fertig
9	Ring Indicator(RI)	Klingel

Tabelle 4.1 Beschreibung einer 9 poligen RS-232 Schnittstelle

In meiner Arbeit werden die TxD (Pin 3), RxD (Pin2) und GND (Pin5) zwischen den ADAM Modulen 4520I und dem Computer verdrahtet. Für eine praktische Nutzung muss der Sender vorher wissen, ob der Empfänger bereits empfangsbereit ist. Die Geschwindigkeit vom Sender ist größer als die Geschwindigkeit vom Empfänger, daher sind sie nicht synchron. Weil die Datenübertragung asynchron arbeitet, um keine Verluste bei der kontinuierlichen Datenübermittlung zu erhalten, benötigt man eine Flußsteuerung, damit der Empfänger, der mit dem Einlesen der Daten nicht nachkommt, dem Sender Einhalt gebieten kann. Gängige Varianten sind Xon/Xoff Flow Control (Software Handshake) und CTS/RTS Flow Control (Hardware Handshake). [mikrocontrollen.net] Das Software Handshake wird durch in die Datenübertragung eingeführte Zeichen gesteuert. Das Hardware Handshake wird durch Signalpegel auf zugehörigen Schnittstellenleitungen realisiert. Durch die Software Handshake und das Hardware Handshake kann der Empfänger den Sender kontrollieren, um die Daten richtig zu verbinden. Das ist die Idee der Komponente „myCom“. Die Komponente existiert bereits und kann direkt in meinem Programm benutzt werden.

Nach diesem Ablauf wird die serielle Schnittstelle zwischen Computer und ADAM Modulen verbunden.

4.2.2 Datenauslesung

Nach der Verbindung des Computers mit den ADAM Modulen können die Daten von den ADAM Modulen ausgelesen werden. Es wird die Komponente „Timer“ hinzugefügt, um jede Sekunde die Daten von den ADAM Modulen automatisch auszulesen.

4.2.3 Datenberechnung

Es wurde nur auf Spannungssignale von ADAM Modulen zugegriffen. Für die Arbeit werden sieben Werte sowohl Spannungen als auch Temperaturen ausgelesen, deshalb müssen den Parametern eigene Funktionen zugeordnet werden.

Für Kanal-Nr.5 bis 8 des ADAM 4017 Moduls, das heißt, folgende Funktion wird für den zugriff verwendet:

```
function TmyADAM.getwert(index:byte):extended;
```

Für Kanal-Nr.1 bis 3 des ADAM4017 Moduls, das heißt, für die drei Temperaturparameter, müssen die ausgelesene Parameter berechnet werden, also Spannung in Temperaturen umgewandelt werden. Für die Berechnung wird folgende Funktion verwendet:

```
function TmyADAM.getTemp(index:byte):string;

function TmyADAM.getTemperatur(x:extended):string;
const B=4540;
const RV=100E3;
const Ri=5E6;
const Uref=5;
var Rn,a: extended;
begin
result:='?';
if (x>0) and (x<(Uref-1)) then begin
  RN:=x/(((Uref-x)/RV)-(x/Ri));
  // Berechnung der Temperatur !!!
  a:=(B*298.15)/(B+ln(RN/RV)*298.15);
  result:=floattostrF(a-298.15,ffixed,10,2)+ ' [°C]';
end;
```

Die Daten wurden von 8 „Panel“ gezeigt. Abbildung 4.2 zeigt die Oberflächenkomponenten, die im ersten Programm benutzt werden.

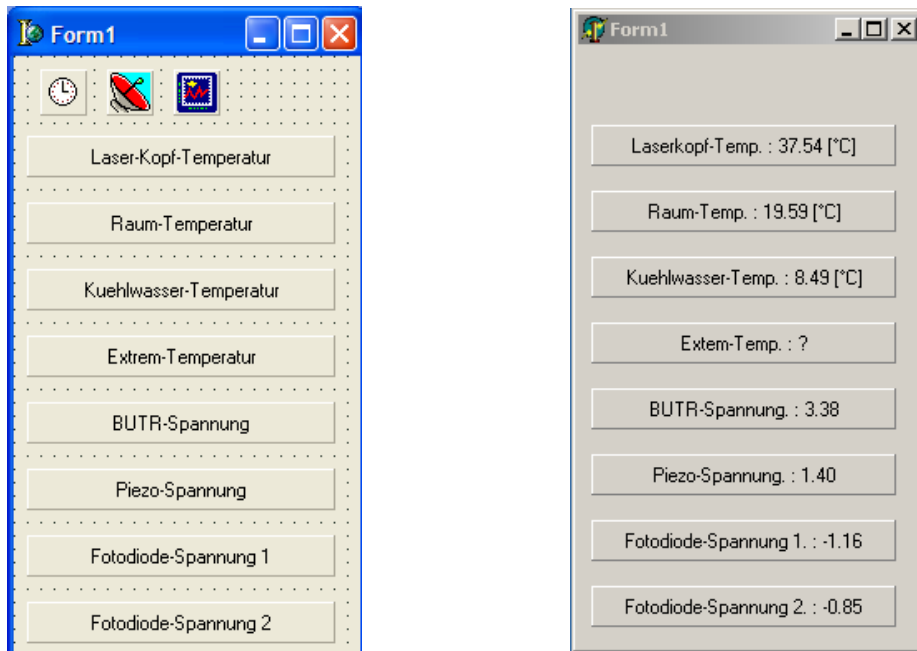


Abbildung 4.2 Oberflächenkomponenten des ersten Programms

4.2.4 Datenspeicherung

Außerdem muss die Datenspeicherung berücksichtigt werden. Die Daten werden als .log Daten auf den Arbeitsplatz des Computers gespeichert. Die Daten können durch drei Prozeduren (openfile, savefile und writeLogfile) und einer Funktion (getDaten) als Public-Deklarationen gespeichert werden.

```
procedure openfile(filename:string);
procedure savefile(filename:string);
procedure writeLogfile(identifier, parameter, data:string);
function getData(kennung, Parameter, name:string):Txyextended;
```

Die Struktur der gespeicherten Daten ist in Abbildung 4.3 dargestellt und hat von links nach folgende Struktur: Datum+Uhrzeit+SDR+P0+die ausgelesenen Parameter.

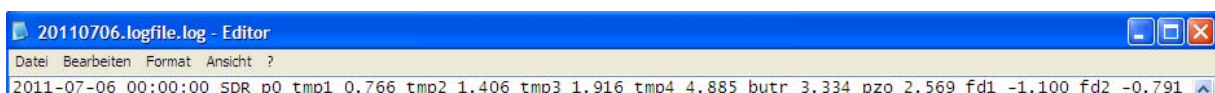


Abbildung 4.3 Struktur der gespeicherten Daten

4.2.5 Grafikerzeugung

Zur graphischen Darstellung der Logdates, wurde nach der Datenspeicherung ein neues Programm programmiert. Zur Laserüberwachung werden sieben Parameter gemessen: BUTR-Spannung, Piezo- Spannung, Fotodiode Spannung 1 und 2 (Spannungsparameter) sowie Laser-Kopf-temperatur, Raumtemperatur und Kühlwassertemperatur (Temperaturparameter) (siehe 4.2.3) und andere drei Werte der Temperaturen zur Temperaturgruppe (Umwandlung

von Spannung und auf Temperatur siehe 4.2.3). Der Wertebereich der gemessenen Parameter umfasst Spannungen zwischen -1,5 V und 5 V sowie Temperaturen zwischen 5°C und 40°C. Die Daten werden von 0 Uhr bis 24 Uhr (UT) geplottet. Beispiel ist in Abbildung 4.4 dargestellt.

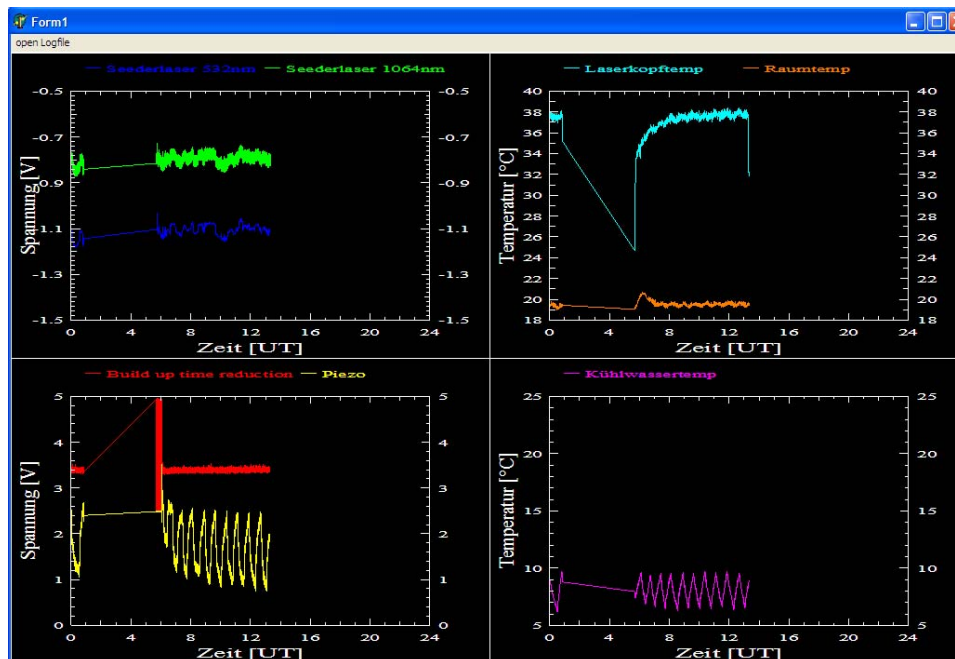


Abbildung 4.4 Grafik der erfassten Parametern am 06.07.2011

4.3 Software-Test

Nach der Programmierung mit Delphi 6 erfolgt der Programmdurchlauf. Es entstanden zwei Fehlermeldungen.

4.3.1 Fehlermeldung : „ zu wenig Arbeitsspeicher“.

Wenn das Programm über drei Stunden gelaufen ist, gab es die Fehlermeldung: zu wenig Arbeitsspeicher. Im Windows Task-Manager war zu sehen, dass bei der Systemleistung die Auslagerungsdatei immer größer wurde. Natürlich steht immer weniger Arbeitsspeicherplatz zur Verfügung. Das Problem liegt in der Speicherung (Procedure writeLogFile). Durch das Freigeben „file Content“ wird das Programm fortgesetzt und die Auslagerungsdatei wächst nicht mehr weiter an.

4.3.2 Fehlermeldung : „EConvertError-Exception“.

Nach längerer Programmausführung wird das Programm mit der Fehlermeldung „ EConvertError“ beendet. Diese Fehlermeldung entsteht, wenn Strtofloat einen bestimmten String in einen Gleitkommawert konvertiert. Falls „String“ keine gültigen Werte enthält, wird eine EConverError-Exception ausgelöst. Diese Exception kann mit Hilfe einer try...except-Anweisung behandelt werden. In einer try...except-Anweisung werden zuerst die

Programmzeilen in einer Anweisungsliste ausgeführt. Werden dabei keine Exceptions ausgelöst, wird der ExceptionBlock ignoriert und die Steuerung an den nächsten Programmteil übergeben. Eine Exception-Behandlungsroutine hat „*on Bezeichner.Typ* do als *Anweisung*. Hier ist *Bezeichner*. optional und kann ein beliebiger Bezeichner sein. *Typ* ist ein für die Exception verwendeter Typ, und *Anweisung* ist eine beliebige Anweisung [Delphi-Hilfe]. Mit dieser Methode wird dieses Problem erfolgreich gelöst. Die Behandlungsroutine ist in dem folgenden Quellcode definiert.

```
procedure TmyADAM.readDaten(astring:string);
var dummy:string;
var errorflag:boolean;
begin
  errorflag:=FALSE;
  dummy:=copy(astring,2,length(astring));
  try
    _rawdata[1]:=strtofloat(copy(dummy,1,7));
    _rawdata[2]:=strtofloat(copy(dummy,8,7));
    _rawdata[3]:=strtofloat(copy(dummy,15,7));
    _rawdata[4]:=strtofloat(copy(dummy,22,7));
    _rawdata[5]:=strtofloat(copy(dummy,29,7));
    _rawdata[6]:=strtofloat(copy(dummy,36,7));
    _rawdata[7]:=strtofloat(copy(dummy,43,7));
    _rawdata[8]:=strtofloat(copy(dummy,50,7));
  except
    on Econvertererror do errorflag:=TRUE;
  end;
```

Nach erfolgreichem Software-Test kann das Programm ohne Fehler durchlaufen.

5. Signal- und Datenverarbeitung

Nach der Fertigung der Laserüberwachung des RMR-Lidars werden die Signale- und Daten verarbeitet. In diesem Kapitel werden die Daten analysiert und verschiedene Abbildungen dargestellt.

5.1 Idee und Programmablaufplan

In der Arbeit wurden die Daten mit MATLAB analysiert. Als eine kommerzielle Software ist MATLAB der Firma The Mathworks zur guten Lösung diverser mathematischer Probleme und zur grafischen Darstellung der Messergebnisse geeignet. Im Vergleich zu Delphi 6 ist es mit MATLAB einfacher, die Daten grafisch darzustellen. Gleichzeitig ist MATLAB mir bekannter. Deshalb wurden die Daten in der Arbeit mit MATLAB verarbeitet. Mit MATLAB können einfache Berechnungen von Matrizen durchgeführt und komplizierte Programme entwickelt werden. Der Quellcode der Benutzeroberfläche wird als sogenannte „M-Files“ bezeichnet. M-Files sind ASCII –Files und werden mit einer Syntax geschrieben. Sie bieten einen Kommandozeileninterpreter und die Visualisierungsmöglichkeit. Außerdem kann die Implementierung eigener Algorithmen mit M-Files realisiert werden. Deswegen wurden M-Files in der Arbeit verwendet.

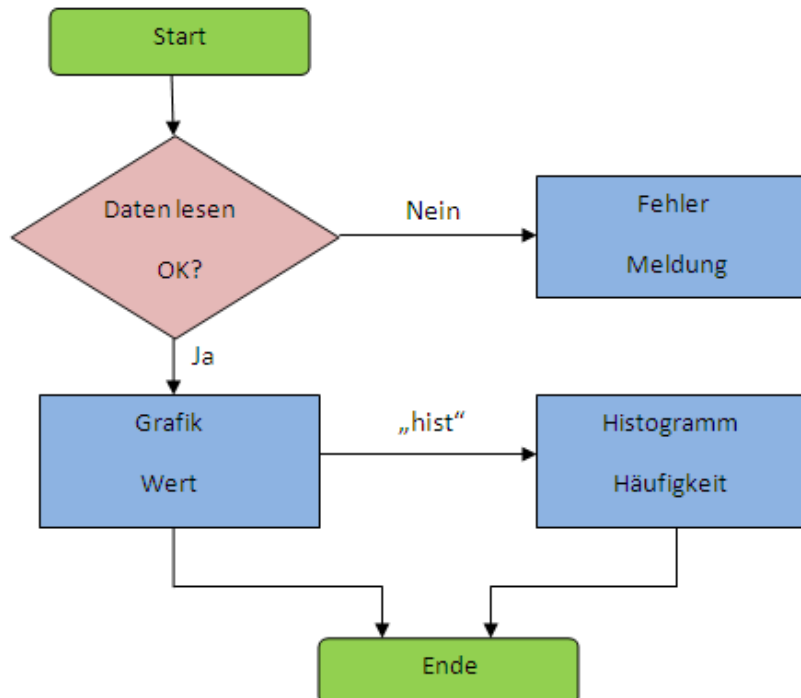


Abbildung 5.1 Ablaufplan der grafischen Darstellung mit MATLAB

Abbildung 5.1 zeigt den Ablaufplan der grafischen Darstellung mit MATLAB. Das Prinzip der grafischen Darstellung mit MATLAB ist genauso wie mit Delphi 6. Es besteht aus drei Schritten. Erstens: Auf die gespeicherten Daten „log“-Edition wurde zuerst mit MATLAB zugegriffen und die acht Artikel tmp1, tmp2, tmp3, tmp4, butr, pzo, df1, df2 aus der .log-Edition wurden als sogenannte dt1, dt2, dt3, dt4, db, dp, df1, df2 Variablen bei MATLAB definiert. Zweitens: dt1, dt2, dt3 sowie dt4 wurden nach der selben Gleichung (siehe 4.2.3) von Spannungswerten in Temperaturwerte umgewandelt. Drittens: Alle Daten wurden später mit dem Befehl „Plot“ in der grafischen Benutzeroberfläche dargestellt. In der Arbeit sind drei verschiedene Grafikgruppen dargestellt.

- Ein Plot in einer Figur

Je gespeicherte Datei wird mit dem Befehl **Plot** eine eigene Abbildung erzeugt. Die Werte werden zwei Gruppen (Spannung und Temperatur) zugeordnet.

- Mehrere Plots in einer Figur

Um die Zusammenhänge zwischen verschiedenen Parametern (hier: Laser-Kopftemperatur, Kühlwassertemperatur und Piezo-Spannung) deutlich sichtbar zu machen, sollen mehrere Plots in einer Figur gezeigt werden.

- Darstellung eines Histogramms (Häufigkeit)

Mit dem Befehl **hist** kann zum jeweiligen erfassten Parameter ein eigenes Histogramm dargestellt werden. Es geht um die Häufigkeiten der erfassten Parameter.

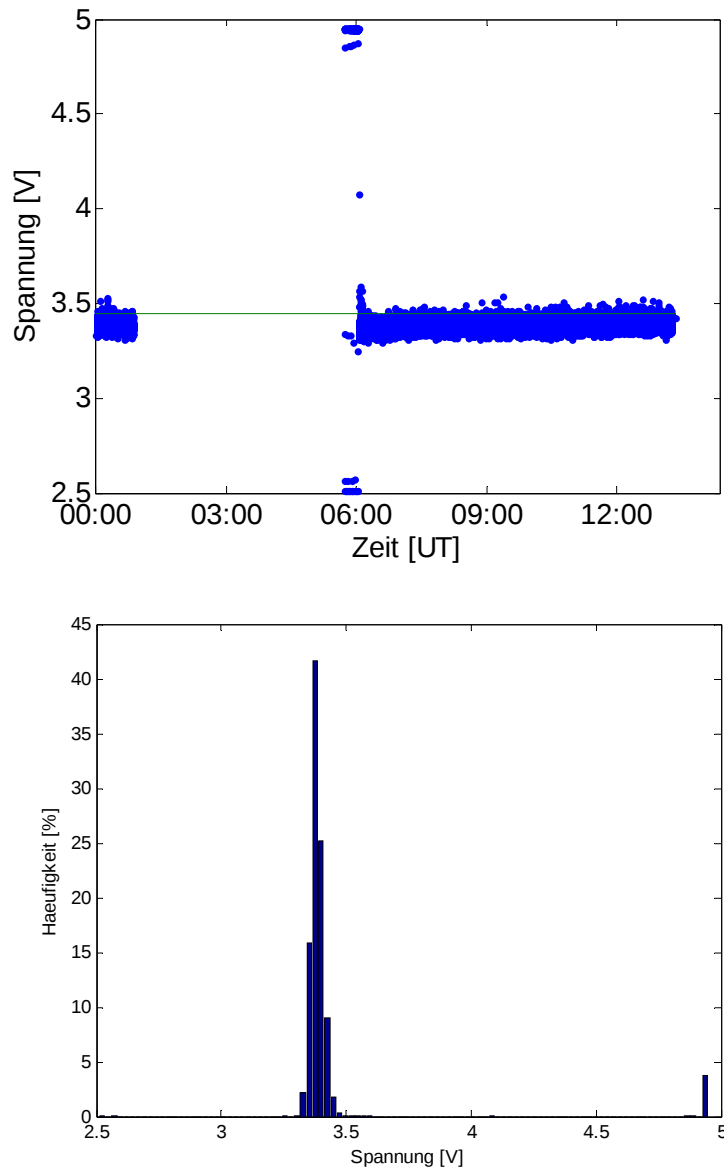
5.2 Analyse der erfassten Parameter

Alle Daten der erfassten Parameter stammen vom 06.07.2011. Die Messung besteht aus zwei Teilen, einer von 0 Uhr bis 0:51 Uhr und der andere von 5:42 Uhr bis 13:20 Uhr. Sie dauerte insgesamt etwa 8,5 Stunden.

5.2.1 Spannungsparameter

1. BUTR (Buld up time reduction) -Spannung sowie ihre Häufigkeit

In Abbildung 5.2 sind die BUTR-Spannung sowie ihre Häufigkeit abgebildet. Alle Werte befinden sich im Bereich zwischen 3,3 V und 3,5 V und der Mittelwert ist 3,45 V. Das Histogramm zeigt deutlich, dass über 90% der Werte der BUTR-Spannung zwischen 3,4 V und 3,45 V liegen, wobei 43% der Werte 3,42 V sind.



5.2 BUTR-Spannung sowie ihre Häufigkeit

2. Piezo-Spannung sowie ihre Häufigkeit

Abbildung 5.3 informiert über die Piezo-Spannung sowie ihre Häufigkeit. Die Piezo-Spannung wurde von 0,8 V bis 2,6 V mit konstanter Amplitude variiert. Sie ändert sich aufgrund thermischer Ausdehnung mit der Zeit. Der typische Zustand ist in Abbildung 5.3 gezeigt. Die Piezo-Spannung wird erst kleiner, dann wieder größer, d.h. sie schwankt mit einer Periode von 50 Minuten um 1,8 V. Der Mittelwert beträgt 2 V. Fast alle Werte liegen im Bereich zwischen 1 V und 2,5 V, wobei der häufigste Wert 1,3 V ist.

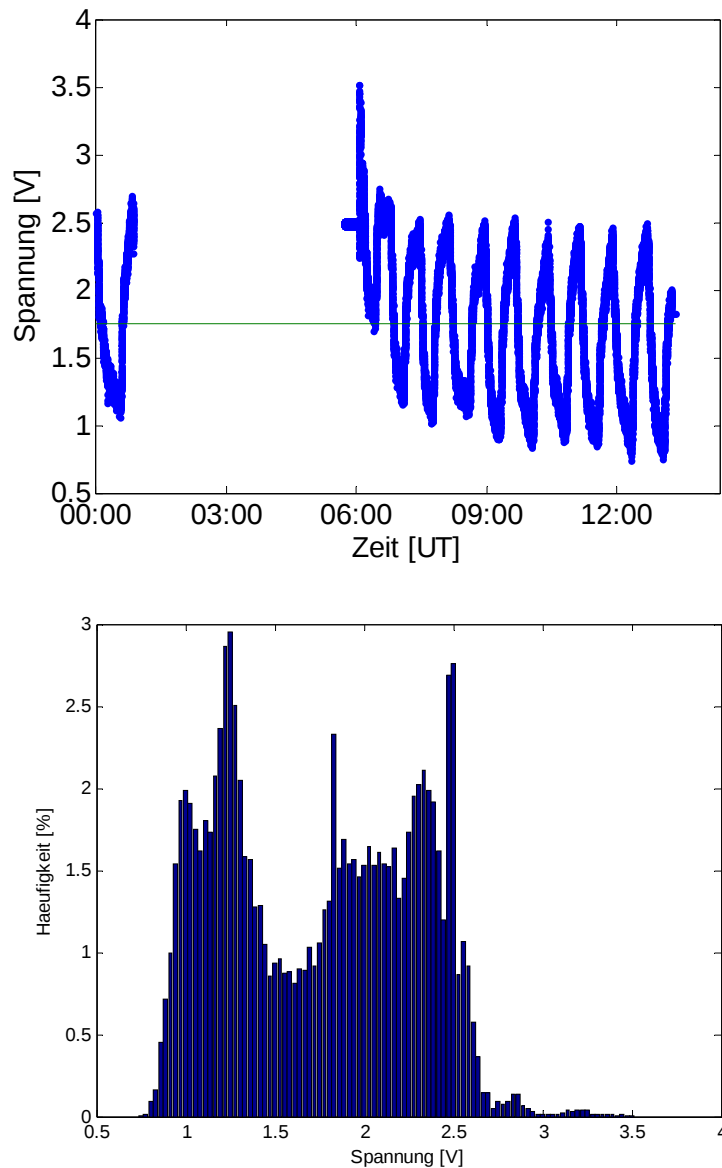


Abbildung 5.3 Piezo-Spannung sowie ihre Häufigkeit

3. Fotodioden-Spannung 1 sowie ihre Häufigkeit

In Abbildung 5.4 (oberes Schaubild) ist das Intensitätsprofil des Seederlasers bei $\lambda=532$ nm für den Zeitraum der Messung gegeben. Wegen des aktiven Tiefpass-Filters sind alle Werte negativ und wurden zuerst in positive Werte umgerechnet. Der maximale Wert ist hier 1,16 V und der minimale Wert beträgt 1,06 V. Die grüne horizontale Linie markiert den Mittelwert von -1,1 V. Das entsprechende Histogramm zeigt, dass der häufigste Wertebereich zwischen -1,1 V und -1,08 liegt.

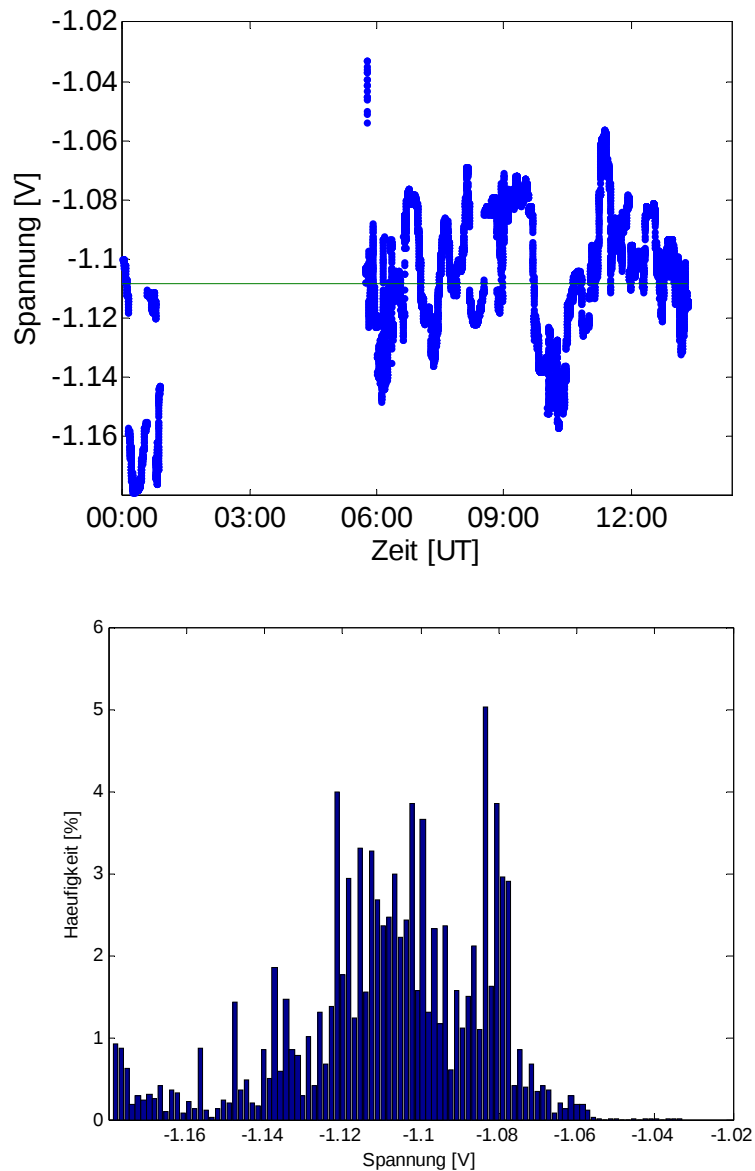


Abbildung 5.4 Fotodioden-Spannung 1 sowie ihre Häufigkeit

4. Fotodioden-Spannung 2 sowie ihre Häufigkeit

In Abbildung 5.5 (oberes Schaubild) ist das Intensitätsprofil des Seederlasers bei $\lambda=1064$ nm für den Zeitraum der Messung gegeben. Wegen des aktiven Tiefpass-Filters sind alle Werte negativ und wurden zuerst in positive Werte umgerechnet. Der maximale Wert ist hier 0,87 V und der minimale Wert beträgt 0,72 V. Die grüne horizontale Linie markiert den Mittelwert von -0,81 V. Das entsprechende Histogramm zeigt, dass der häufigste Wertebereich zwischen -0,85 V und -0,76 liegt.

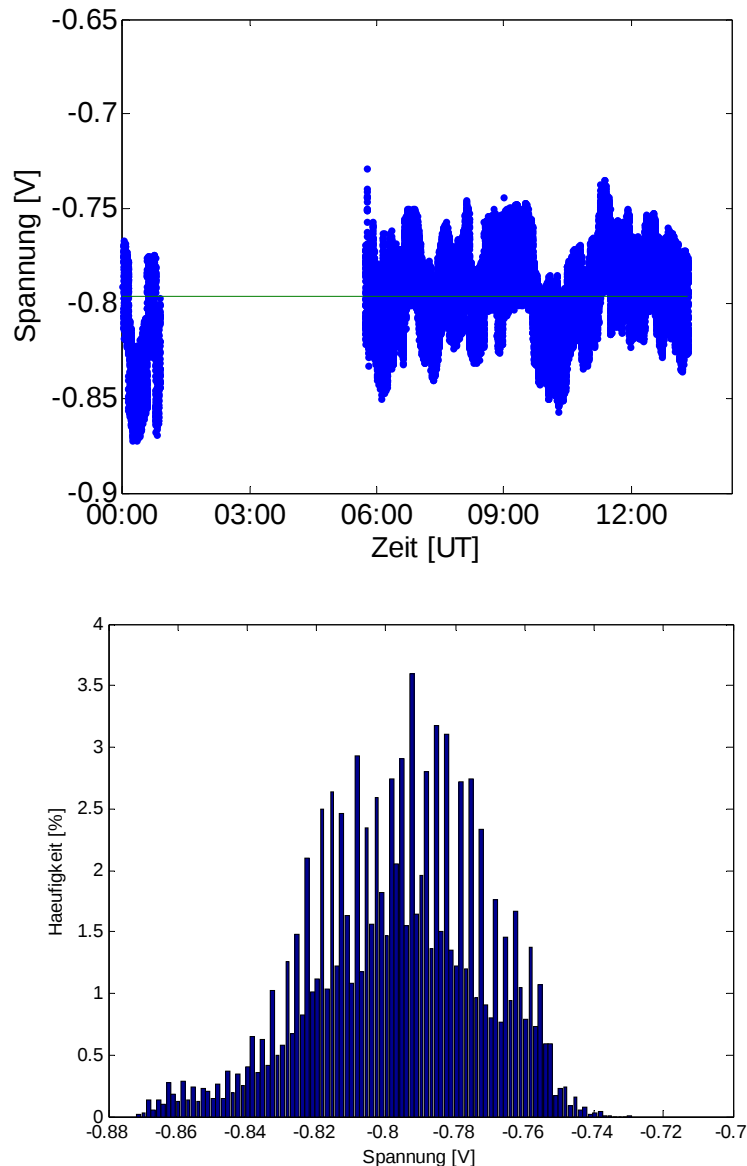


Abbildung 5.5 Fotodioden-Spannung 2 sowie ihre Häufigkeit

5.2.2 Temperaturparameter

1. Laser-Kopftemperatur sowie ihre Häufigkeit

Durch die Abbildung 5.6 (oberes Schaubild) kann man feststellen, dass wenn der Leistungslaser läuft, die Laserkopftemperatur zunimmt. Die Starttemperatur ist etwa 20°C, also ähnlich wie die Raumtemperatur. Sie steigt bis etwa 38°C und stabilisiert sich dort. Dieser Ablauf dauerte circa 2 Stunden. Danach wird die Laser-Kopftemperatur durch den Kühlwasserkreislauf reguliert, der im Temperaturbereich zwischen 36°C und 38°C gehalten wurde, also $\pm 1^\circ\text{C}$ konstant war. Die Laser-Kopftemperatur hat eine zeitliche Schwankung mit der Dauer von etwa 50 Minuten. (siehe 5.3). Der Mittelwert beträgt 37,1°C (grüne Linie). Die Häufigkeit der Laser-Kopftemperatur ist in Abbildung 5.6 (unteres Schaubild) dargestellt.

Dabei fällt auf, dass die Laser-Kopftemperatur zu über 90% der Untersuchung zwischen 36°C und 38°C lag.

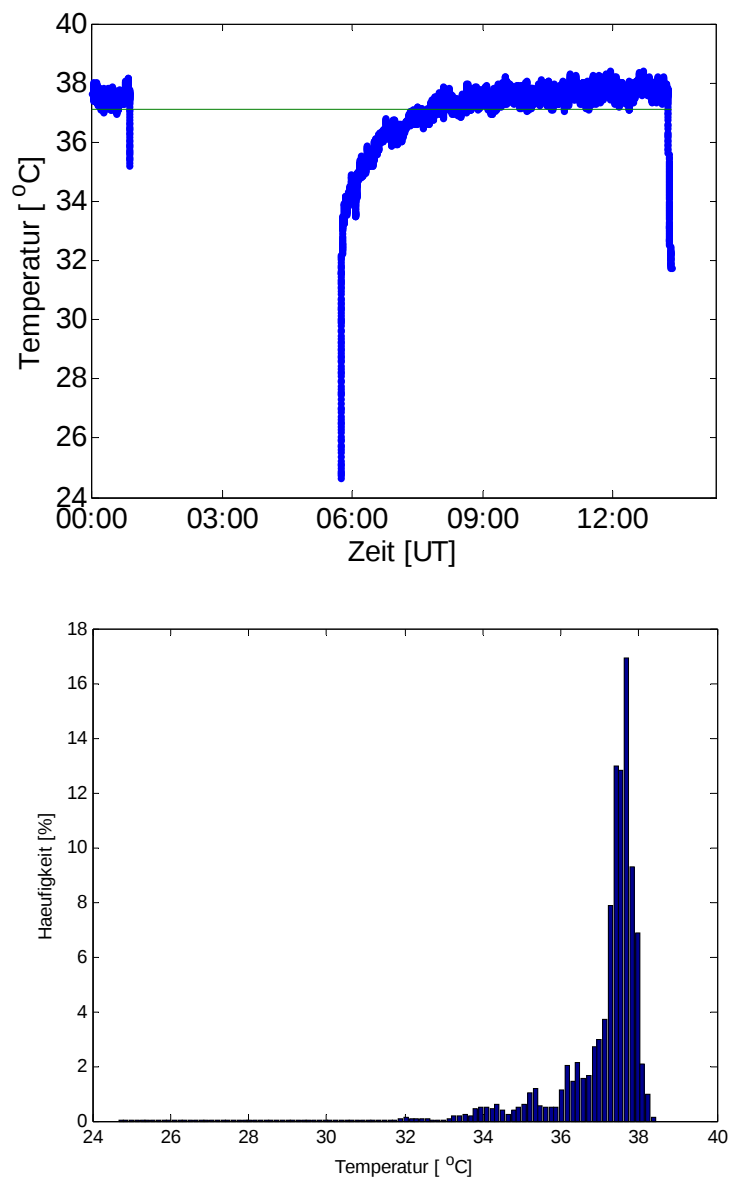


Abbildung 5.6 Laser-Kopftemperatur sowie ihre Häufigkeit

2. Raumtemperatur sowie ihre Häufigkeit

Die Lufttemperatur des Labors wird über den Zeitraum der Messung mit einer Klimaanlage konstant gehalten. Dadurch wird die Raumtemperatur bei etwa 20°C stabilisiert. Abbildung 5.7 zeigt, dass der Raumtemperaturbereich zwischen 19°C und 20,6°C ist. Die Raumtemperatur hat eine periodische Schwankung mit der Periodendauer von etwa 50 Minuten und der Mittelwert ist 19,55°C. Das rechte Schaubild zeigt, dass der häufigste Bereich zwischen 19,3°C und 19,6°C liegt, mit jeweils über 4% Häufigkeit.

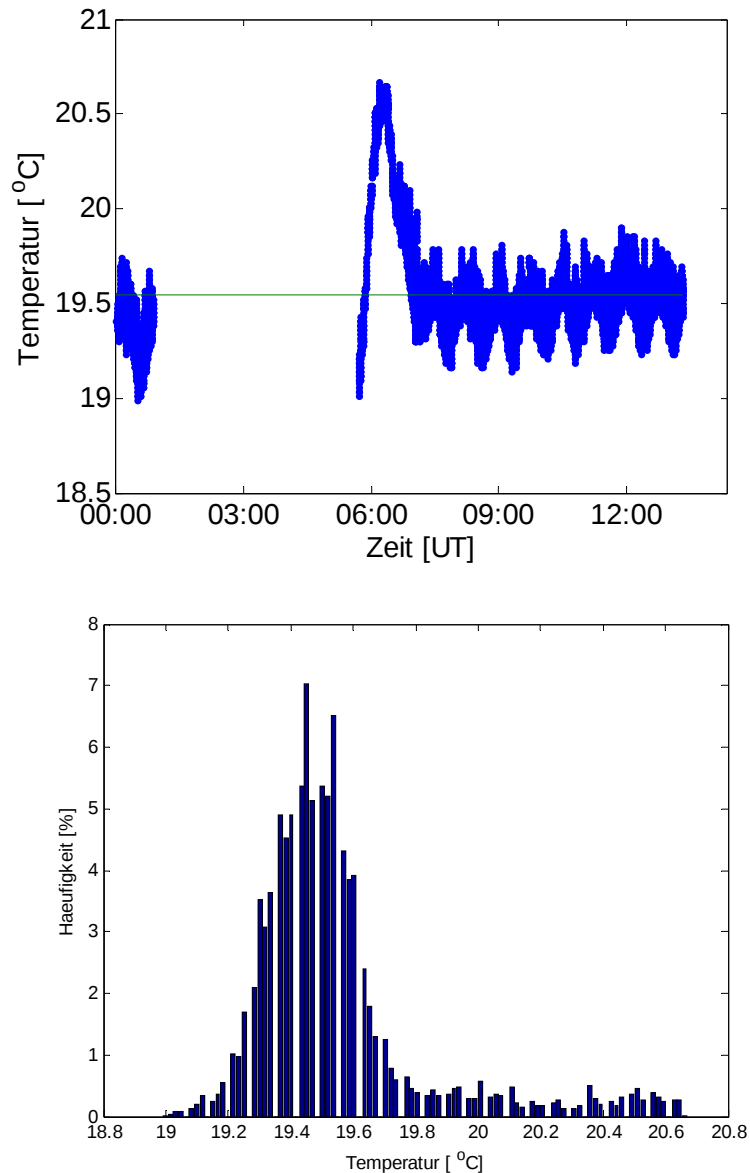


Abbildung 5.7 Raumtemperatur sowie ihre Häufigkeit

3. Kühlwassertemperatur sowie ihre Häufigkeit

Der Verlauf der Kühlwassertemperatur wird in Abbildung 5.8 (oberes Schaubild) dargestellt. Wie vorher erklärt, befindet sich der Laserkopf im Kühlwasserkreislauf. Den Zusammenhang zwischen der Laser-Kopf-temperatur und der Kühlwassertemperatur zeigt 2.3. Zwischen 9:20 Uhr und 10:10 Uhr schwankt die Kühlwassertemperatur ähnlich wie die Raumtemperatur (Periode von 50 Minuten), weil beide die gleiche Kühlanlage benutzen. Das Maximal ist bei rund 6,2°C, das Minimal ist von rund 9,8°C und der Mittelwert bzw. der Sollwert ist rund 8°C. Die Kühlwassertemperatur ist $\pm 2^\circ\text{C}$ konstant. Die Häufigkeit der Kühlwassertemperatur, wie Abbildung 5.8 (unteres Schaubild) zeigt, hat keine großen Variationen (jeweiliger Wert ist ungefähr 1,2%).

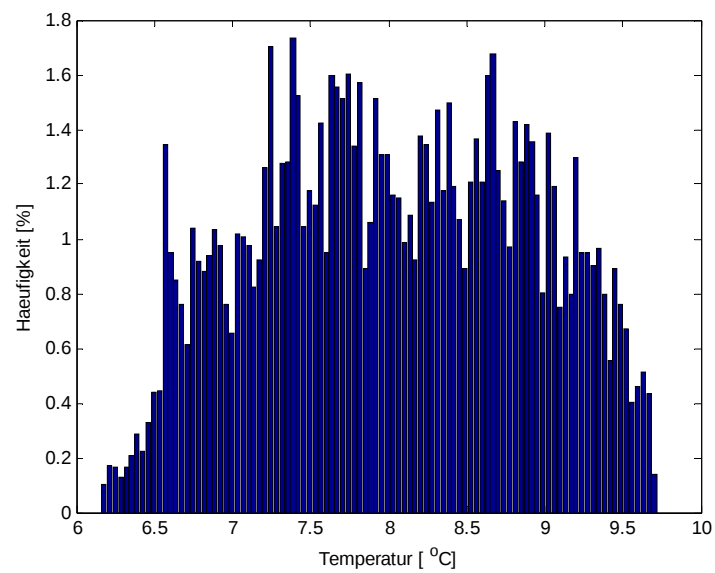
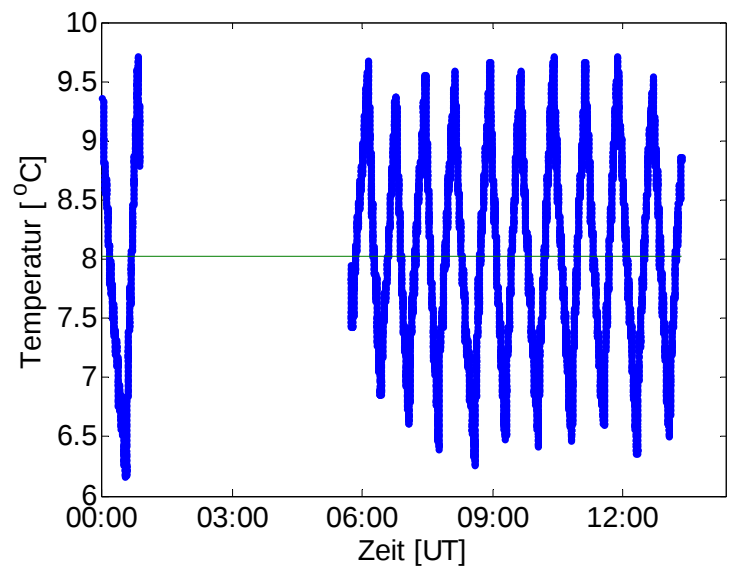


Abbildung 5.8 Kühlwassertemperatur sowie Ihre Häufigkeit

5.3 Zusammenhang zwischen verschiedenen Parametern

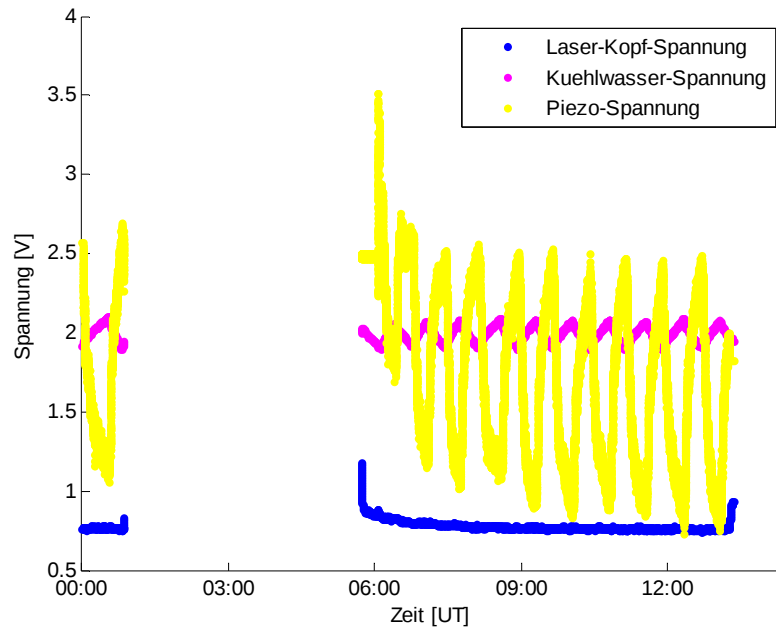


Abbildung 5.9 Zusammenhänge zwischen verschiedenen Parametern

Abbildung 5.9 zeigt den Zusammenhang zwischen der Laserkopf-Spannung (blau), der Kühlwasser-Spannung (rosa) und der Piezo-Spannung (gelb). Für bessere Übersichtlichkeit werden Spannungen ausstellt von Temperaturen dargestellt. Durch den Charakter der NTC 100 und nach deren Gleichungen (3.5-3.11) wird es verdeutlicht: Je größer die Spannung ist, desto niedriger die Temperatur. Weil die Temperaturänderung des Kühlwasser ($3,6^{\circ}\text{C}$) größer als die Temperaturänderung des Laserkopfs (2°C) ist (siehe 5.2.1), hat die Kühlwasser-Spannung eine größere Amplitude. Gleichzeitig haben die Laserkopf-Spannung und die Kühlwasser-Spannung eine gleiche Periodendauer von 50 Minuten zwischen maximalem Wert und minimalem Wert. Die Laser-Kopf-Spannung sollte immer der Kühlwasser-Spannung nachlaufen, d.h. wenn das externe Kühlwasser wärmer wird, wird auch das interne Kühlwasser wärmer. Später wird dann auch der Laserkopf wärmer. Wenn das externe Kühlwasser kälter wird, wird das interne Kühlwasser kälter. Später wird der Laserkopf auch kälter. Durch die Überwachung kann das Kühlsystem reguliert werden.

Die Piezo-Spannung gleicht die Längenänderung des Resonators sofort aus. Die Längenänderung beginnt eventuell später als die Änderung der Laser-Kopf-Spannung wegen der thermischen Leitung.

6. Zusammenfassung und Ausblick

Im Rahmen der Arbeit wurde ein Laserparameterüberwachungssystem für das neue RMR-Lidar am IAP entwickelt und aufgebaut. Das neue Kühlungsborner RMR-Lidar ermöglicht seit 2010 Temperaturmessungen in der mittleren Atmosphäre bei Tag als auch bei Nacht. Die für Lidar-Anwendungen verwendeten Laser müssen wegen der hohen Anforderungen an Leistung und Präzision in teilweise kritischen technischen Grenzbereichen betrieben werden. Um Schäden an den Systemen zu verhindern und die Betriebsparameter optimal einzustellen, wurde in dieser Arbeit ein Überwachungsmodul für die Laserkühlung und die wichtigsten Leistungsparameter entwickelt und aufgebaut. Somit können relevante Überwachungsparameter, wie Spannungsparameter (Fotodioden Spannung 1 und 2, BUTR-Spannung, Piezo-Spannung) sowie Temperaturparameter (Laser-Kopftemperatur, Raumtemperatur, Kühlwassertemperatur) gemessen werden. Das Überwachungssystem basiert auf der Grundlage von ADAM Modulen, die bereits erfolgreich in anderen Bereichen des IAPs (Eisen-Lidar) eingesetzt werden. Im Rahmen dieser Arbeit wurden zwei Platinen (Fotodiodenverstärker und Temperaturmessung) layoutet und gefertigt. Die Platine für den Fotodiodenverstärker misst die Leistung des cw-Lasers und die Platine für die Temperaturmessung misst alle drei Temperaturparameter. Um das Auslesen der überwachten Parameter abhängig von der jeweiligen Aktualisierungsrate und eine Darstellung der Werte auf einem PC zu schaffen, erfolgte die Programmierung mit Delphi 6.

Die Auswertung der Logdateien ermöglicht eine Optimierung der Betriebsparameter wie der Raumtemperatur sowie der Kühlwassertemperatur. Damit wird es in Zukunft möglich sein, eine bessere thermische Stabilität des Lasersystems zu erzielen. Zudem werden die Parameter während der Messung online dargestellt, um ein eventuelles Eingreifen oder ein Abbruch der Messung zu bewirken, falls relevante Parameter vorgegebene Grenzen überschreiten. Durch die Überwachung sowie Auswertung der Laserparameter soll in einer späteren Ausbauhilfe der Laser bei Überschreiten voreingestellter Grenzen automatisch ausgeschaltet werden, um Schäden am System zu verhindern.

Anhang A

A.1 Quellcode vom Programm ADAM

```
program ADAM;
uses
  Forms,
  Unit2 in 'C:\Programme\Borland\Delphi6\Projects\Unit2.pas' {Form1};
{$R *.res}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComDrv32, mycom, myADAM, ExtCtrls, Meta, achsen,
  felder, contLines, plotXy, store, calc, Plots, Xy_ext, Menus, zoom;
type
  TForm1 = class(TForm)
    mycom1: Tmycom;
    Timer1: TTimer;
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    Panel6: TPanel;
    Panel7: TPanel;
    Panel8: TPanel;
    ZoomPopUpMenu1: TZoomPopUpMenu;
    procedure FormCreate(Sender: TObject);
    procedure mycom1Getstring(sender: TObject; var st: string);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private-Deklarationen }
    _myadam:TMyADAM;
    procedure showAdamData(sender:Tobject);
  public
    { Public-Deklarationen }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
var
```

```

inbuff,outbuff:array of byte;
//defined by dynamic array input and output buffers
data_received:array of smallint absolute inbuff;
// receive array of integer, array of store byte coincides with the Inbuff
(*)
begin
_myAdam.readconfig;
//mycom1.SendString('#01'#13);
Timer1.Interval:=1000;//1 Sekunde
Timer1.Enabled:=true;//Timer starten
end;
*)
procedure TForm1.FormCreate(Sender: TObject);//load window event
begin
_myAdam:=TMyAdam.create(self);
_myAdam.com:=mycom1;
_myAdam.Adresse:='01';
_myAdam.onnewdata:=showAdamData;
//Timer1.Enabled:=false;//timer Deaktibieren
mycom1.startchar:='';
mycom1.endechar:='#13';
mycom1.Connect;
//_myAdam.readconfig;
//mycom1.SendString('#01'#13);
Timer1.Interval:=1000;//1 Sekunde
Timer1.Enabled:=true;//Timer starten
end;
procedure TForm1.mycom1Getstring(sender: TObject; var st: string);
begin
// form1.caption:=st;
_myAdam.readADAMDaten(st);
end;
procedure TForm1.showAdamData(sender:TObject);
begin
with _myAdam do begin
panel1.Caption:='Laserkopf-Temp. : '+ Temperatur[1];
panel2.Caption:='Raum-Temp. : '+ Temperatur[2];
panel3.Caption:='Kuehlwasser-Temp. : '+ Temperatur[3];
panel4.Caption:='Extem-Temp. : '+ Temperatur[4];
panel5.Caption:='BUTR-Spannung. : '+floattostrF( Wert[5],ffixed,5,2) ;
panel6.Caption:='Piezo-Spannung. : '+floattostrF(Wert[6],ffixed,5,2);
panel7.Caption:='Fotodiode-Spannung 1. : '+floattostrF(Wert[7],ffixed,5,2);
panel8.Caption:='Fotodiode-Spannung 2. : '+floattostrF(Wert[8],ffixed,5,2);
end;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
//Anweisung
_myADAM.getDaten;
end;

```

end.

A.1.1 Quellcode der Datenübertragung

```
unit mycom;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComDrv32;
type ongetstringevent = procedure (sender:Tobject;var st:shortstring) of object;
type oncheckerrorevent = procedure (sender:Tobject)of object;
type anwendungstyp = (allgemein,Owis,Synchronisation,focus,logger);
type
  Tmycom = class(TCommPortDriver)
private
  { Private-Deklarationen }
  _check :boolean;
    {Prüfung der Checksumme ein aus}
  _error :boolean;
    {zeigt Fehler bei der Übertragung an}
  _stringaz :shortstring;
    {Stringanfangszeichen}
  _stringez :shortstring;
    {Stringendeezeichen}
  _checksumme :byte;
    {Checksumme über alles}
  _empfang :shortstring;
    {aktueller Empfangsstring}
  _lastsendstring :string; {letzter
gesendeter String}
  _anwendung :anwendungstyp;
  Foncheckerror :oncheckerrorevent;
  Fgetstringevent :ongetstringevent;
  function check(var s:shortstring):char; {Checksumme bilden}
  procedure setanwendung(x:anwendungstyp);
  function getstatus:boolean;
protected
  procedure achieveData(FTemplnBuffer:Pointer;nread:uint);override;
public
  { Public-Deklarationen }
  constructor create(AOwner: TComponent); override;
  destructor destroy;override;
  function INFreeSpace: word;
  function SendmyString(s: shortstring ): boolean;
  function CTS_Status:boolean;
  function DSR_Status:boolean;
  function checkRMRPuls:byte; (* Funktion für Kontrolle der Synchronisation Metall-Lidar*)
  property Status :boolean read getstatus;
  property startchar :shortstring read _stringaz write _stringaz;
```

```

property endechar :shortstring read _stringez write _stringez;
published
{ Published-Deklarationen }
property checksumme:boolean read _check write _check;
property Anwendung:anwendungstyp read _anwendung write setanwendung;
property OnGetString:ongetstringevent read Fgetstringevent write Fgetstringevent;
property OnCheckerror:oncheckerrorevent read Foncheckerror write Foncheckerror;
end;
implementation
constructor Tmycom.create(Aowner:Tcomponent);
begin
inherited create(Aowner);
_check                :=false;    {keine Prüfung der Checksumme}
_error                :=false;
_stringaz              :='';      {Stringanfangszeichen}
_stringez              :='';      {Stringendezeichen}
_empfang              :='';      {aktueller Empfangsstring}
_lastsendstring        :='';
_checksumme            :=0;
end;
destructor Tmycom.destroy;
begin
inherited destroy;
end;
procedure Tmycom.setanwendung(x:anwendungstyp);
begin
case x of
    focus          :begin _stringaz:='@';    _stringez:=#13;end;
    logger          :begin _stringaz:='$'; _stringez:=#13;end;
    synchronisation :begin _stringaz:='<';    _stringez:='>';end;
    owis            :begin _stringaz:='<';    _stringez:='>';end;
    allgemein        :begin _stringaz:='';      _stringez:='';end;
end;
_anwendung:=x;
end;
function Tmycom.getstatus:boolean;
begin
if not _error then result:=true else result:=false;
_error:=false;
end;
function Tmycom.CTS_Status:boolean;
{$IFDEF VER100}
var ModemStat:integer;
{$ELSE}
var ModemStat:cardinal;
{$ENDIF}
begin
GetCommModemStatus(comHandle,ModemStat);
if ModemStat and MS_CTS_ON=ms_cts_on then result:=true else result:=false;
end;

```

```

function Tmycom.DSR_Status:boolean;
{$IFDEF VER100}
var ModemStat:integer;
{$ELSE}
var ModemStat:cardinal;
{$ENDIF}
begin
GetCommModemStatus(comHandle,ModemStat);
if ModemStat and MS_DSR_ON=ms_DSR_on then result:=true else result:=false;
end;
function Tmycom.checkRMRPuls:byte;
// Funktion nur für die Synchronisationselektronik zu gebrauchen !!!
    Funktion wertet die Statusleitungen der RS232 aus, um festzustellen , ob das
    RMR.Lidar in der Messzeit vom Ca- oder K-Lidar gefeuert hat.
    result=0   - RMR hat nicht gestört
    result=1   - RMR hat Ca gestört
    result=2   - RMR hat K gestört
    result=3   - RMR hat beide gestört
*)
function Tmycom.checkRMRPuls:byte;
{$IFDEF VER100}
var ModemStat:integer;
{$ELSE}
var ModemStat:cardinal;
{$ENDIF}
var K,Ca:byte;
begin
GetCommModemStatus(comHandle,ModemStat);
if ModemStat and MS_RING_ON=ms_RING_on then K:=0 else K:=1;
if ModemStat and MS_DSR_ON=ms_DSR_on then Ca:=0 else Ca:=1;
result:=3-(k+2*Ca);
end;
{ Returns the input buffer free space or 65535 if not connected }
function Tmycom.INFreeSpace: word;
var stat: TCOMSTAT;
    errs: DWORD;
begin
    if not Connected then
        Result := 65535
    else
        begin
            ClearCommError( FComPortHandle, errs, @stat );
            Result := FComPortINBufSize - stat.cbINQue;
        end;
    end;
end;
function Tmycom.SendmyString( s: shortstring ): boolean;
var checksumme:char;
var temp:string;
begin
temp:=s;

```

```

if not connected then connect;
if _check then
begin
checksumme:=check(s);
s:=_stringaz+s+checksumme+_stringez;
end
else
begin
if _stringaz<>' ' then s:=_stringaz+s;
if _stringez<>' ' then s:=s+_stringez;
end;
result:=sendstring(s);
if result then _lastsendstring:=temp;
end;
{Die Checksumme wird ohne Anfangs- und Endzeichen gebildet}
function Tmycom.check(var s:shortstring):char;
const maxzeichen=10000;
type feld=array[1..maxzeichen]of byte;
var n,i:integer;
var summe:byte;
begin
n:=length(s);
if n>maxzeichen then n:=maxzeichen;
summe:=0;
for i:=1 to n do
begin
{$R-}
INC(summe,ord(s[i]));
{$R+}
end;
result:=char(summe);
end;
procedure TMyCom.achieveData(FTempInBuffer:pointer;nread:uint);      {String zusammensetzen }
var i:integer;
var bytepnt:^byte;
var checksumme:char;          { empfangene Checksumme}
var zeichen:char;
begin
bytepnt:=FTempInBuffer;
checksumme:=char(32);          { einfach auf einen Wert setzen }
if (length(_empfang)=0) and (_anwendung=focus) then _empfang:='@';
if (length(_empfang)=0) and (_anwendung=logger) then _empfang:='$';
for i:=1 to nread do
begin
zeichen:=char(bytepnt^);
if (zeichen<>_stringEz) then
begin
_empfang:=_empfang+zeichen;
checksumme:=zeichen;          {Zeichen vor dem Stringendezeichen -> Checksumme}
end
end

```

```

else
    begin
    if _check then
        begin
            setlength(_empfang,length(_empfang)-1); {Checksumme im String löschen}
            if check(_empfang)<>checksumme then if Assigned(Foncheckerror) then Foncheckerror(self);
            end;
            if (length(_empfang)>0) then
                if _empfang[1]=_stringAz then
                    begin
                        delete(_empfang,1,1);
                        if Assigned(Fgetstringevent) then Fgetstringevent(self,_empfang);
                    end
                else
                    if Assigned(Fgetstringevent) then Fgetstringevent(self,_empfang);
                end;
            _empfang:="";
            end;
            inc(bytepnt);
        end;
    end;
end.
end.

```

A.1.2 Quellcode der Datenauslesung und Datenberechnung

```

unit myADAM;
interface
// Testversion
uses
    Windows, Messages, SysUtils, Classes, MyComp,mycom,rmrlogfile;
type TrawData=array[0..8] of extended;
type
    TmyADAM = class(Tmycomponent)
    private
        { Private-Deklarationen }
        _com : TMYCom;
        _adr : String;
        _rawdata : TRawData;
        _logfile : TRMRLogfile;
        _lastCMD : string;
        _Fonnewdata : TNotifyEvent;
        procedure readDaten(aString:string);
        function getwert(index:byte):extended;
        function getTemp(index:byte):string;
        function getTemperatur(x:extended):string;
    protected
        { Protected-Deklarationen }
    public
        { Public-Deklarationen }
    end;
end.

```

```

procedure initModul;
procedure getDaten;
procedure readConfig;
procedure readADAMDaten(astring:string);
property Adresse : string read _adr write _adr;
property Wert[index:byte] : extended read getWert;
property Temperatur[index:byte]:string read getTemp;
published
{ Published-Deklarationen }
property com:TMYcom read _com write _com;
property onnewdata : TNotifyevent read _FonnewData write _FonnewData;
end;
implementation
procedure TmyADAM.readADAMDaten(astring:string);
begin
// Auswertung der Antworten vom Modul auf unterschiedliche Befehle !!
// > 1.050 2.080 5.003 7.100 ....
// pos(' ',astring)
// x:=copy(astring,3,length(astring))
// strtfloat
if _logfile=nil then begin
_logfile:=TRMRLogfile.create(nil);
_logfile.path:='c:\adamlog\';
// _logfile.path:='d:\output\RMRLaser\';
end;
if _lastCMD='DATA' then readDaten(astring);
end;
procedure TmyADAM.getDaten;
begin
if _com <> nil then
_com.SendString('#'+_adr+#13);
_lastCMD:='DATA';
end;
procedure TmyADAM.readconfig;
begin
if _com <> nil then
_com.SendString('$'+_adr+'2'+#13);
_lastCMD:='CONFIG';
end;
function TmyADAM.getwert(index:byte):extended;
begin
result:=_rawdata[index];
end;
function TmyADAM.getTemp(index:byte):string;
begin
if index=1 then result:=getTemperatur(_rawdata[1])
else if index=2 then result:=getTemperatur(_rawdata[2])
else if index=3 then result:=getTemperatur(_rawdata[3])
else if index=4 then result:=getTemperatur(_rawdata[4])
else result:='?';
end;

```

```

end;
function TmyADAM.getTemperatur(x:extended):string;
const B=4540;
const RV=100E3;
const Ri=5E6;
const Uref=5;
var Rn,a: extended;
begin
result:='?';
if (x>0) and (x<(Uref-1)) then begin
  RN:=x/(((Uref-x)/RV)-(x/Ri));
  // Berechnung der Temperatur !!!
  a:=(B*298.15)/(B+ln(RN/RV)*298.15);
  result:=floattostrF(a-298.15,ffixed,10,2)+ ' [°C]';
end;
end;
procedure TmyADAM.initModul;
begin
if _com <> nil then
  _com.SendString('$'+_adr+'01020304'+#13);
end;
procedure TmyADAM.readDaten(astring:string);
var dummy:string;
var errorflag:boolean;
begin
errorflag:=FALSE;
dummy:=copy(astring,2,length(astring));
try
  _rawdata[1]:=strtofloat(copy(dummy,1,7));
  _rawdata[2]:=strtofloat(copy(dummy,8,7));
  _rawdata[3]:=strtofloat(copy(dummy,15,7));
  _rawdata[4]:=strtofloat(copy(dummy,22,7));
  _rawdata[5]:=strtofloat(copy(dummy,29,7));
  _rawdata[6]:=strtofloat(copy(dummy,36,7));
  _rawdata[7]:=strtofloat(copy(dummy,43,7));
  _rawdata[8]:=strtofloat(copy(dummy,50,7));
except
  on Econverterror do errorflag:=TRUE;
end;
// Datum setzen
if errorflag then exit;
_rawdata[0]:=now;
if assigned(onnewdata) then _FonnewData(self);
_logfile.writeLogfile('SDR','p0','tmp1 '+floattostrF(_rawdata[1],ffixed,7,3)+' ' +
'tmp2 '+floattostrF(_rawdata[2],ffixed,7,3)+' ' + 'tmp3 '+floattostrF(_rawdata[3],
ffixed,7,3)+' ' + 'tmp4 '+floattostrF(_rawdata[4],ffixed,7,3)+' ' + 'butr ' +
floattostrF(_rawdata[5],ffixed,7,3)+' ' + 'pzo '+floattostrF(_rawdata[6],ffixed,
7,3)+' ' + 'fd1 '+floattostrF(_rawdata[7],ffixed,7,3)+' ' + 'fd2 '+floattostrF
(_rawdata[8],ffixed,7,3));
end;

```

end.

A.1.3 Quellcode der Datenspeicherung

```
unit rmrlogfile;
interface
uses
  Windows, Messages, SysUtils, DateUtils, Classes, mycomp, xy_ext;
type TDaten = record
  zeit : TDateTime;
  wert : extended;
end;

type
  Trmrlogfile = class(Tmycomponent)
  private
    { Private-Deklarationen }
    _Logfile:TStringlist;
    _filename:string;
    _counter:integer;
    _feld:Txyextended;
    _path:string;
  function getlogvalue(logstr,kennung,Parameter,name:string):TDaten;
  function getcount:integer;
  protected
    { Protected-Deklarationen }
  public
    { Public-Deklarationen }
    procedure openfile(filename:string);
    procedure savefile(filename:string);
    procedure writeLogfile(identifier, parameter, data:string);
    function getDate(kennung,Parameter,name:string):Txyextended;
    property count:integer read _counter;
    property daten:Txyextended read _Feld;
    property path:string read _path write _path;
  published
    { Published-Deklarationen }
  end;
implementation
procedure TRMRlogfile.openfile(filename:string);
begin
  if _Logfile = nil then _logfile:=TStringlist.Create;
  if filename <> _filename then _logfile.Clear;
  _logfile.LoadFromFile(filename);
  _Filename:=filename;
end;

procedure TRMRlogfile.writeLogfile(identifier, parameter, data:string);
var DateNow: string;
```

```

var filecontent:TStringlist;
begin
  data:=identifier+' '+parameter+' '+data;
  filecontent:= TStringList.create;
  DateNow:=FormatDateTime('yyyymmdd',Date);
  if FileExists(_path+DateNow+'.logfile.log') then
    filecontent.LoadFromFile(_path+DateNow+'.logfile.log');
    filecontent.Add(FormatDateTime('yyyy-mm-dd hh:nn:ss', now)+' '+Data);
    filecontent.SaveToFile(_path+DateNow+'.logfile.log');
    filecontent.Free;
end;
procedure TRMRlogfile.savefile(filename:string);
begin
  _logfile.SaveToFile(filename);
  _Filename:=filename;
end;
function TRMRlogfile.getlogvalue(logstr,kennung,Parameter,name:string):TDaten;
var datum:string;
var dummy :string;
var time: TDateTime;
var year,month,day,hour,minute,second:string;
begin
  result.wert:=0;
  result.zeit:=0;
  datum:=copy(logstr,1,19);
  if (pos(kennung,logstr)<>0) and (pos(Parameter,logstr)<>0) then begin
    dummy:=copy(logstr,28,length(logstr));
    dummy:=copy(dummy,pos(name+' ',dummy),length(dummy));
    dummy:=copy(dummy,length(name)+1,length(dummy));
    repeat
      dummy:=copy(dummy,2,length(dummy));
    until copy(dummy,1,1)<>' ';
    if pos(' ',dummy)<>0 then
      dummy:=copy(dummy,1,pos(' ',dummy)-1);
    result.wert:=strtofloat(dummy);
  // Datum lesen
  year:=copy(logstr,1,4);
  month:=copy(logstr,6,2);
  day:=copy(logstr,9,2);
  hour:=copy(logstr,12,2);
  minute:=copy(logstr,15,2);
  second:=copy(logstr,18,2);
  result.zeit:=EncodeDateTime(StrToInt(year),StrToInt(month),StrToInt(day),StrToInt(hour),StrToInt(minute),StrToInt(second),0);
  end;
end;
function TRMRLogfile.getcount:integer;
begin
  result:=_logfile.Count;
end;

```

```

function TRMRLogfile.getDaten(kennung,Parameter,name:string):Txyextended;
var i:integer;
var data:TDaten;
begin
  if _feld=nil then _feld:=Txyextended.create(nil);
  _Feld.N:=0;
  if (_logfile <> nil) and (_logfile.Count > 0) then begin
    for i:=0 to _logfile.Count -1 do begin
      data:=getlogvalue(_logfile.Strings[i],kennung,Parameter,name);
      if data.zeit<>0 then
        _feld.appendXy(24*frac(data.zeit),data.wert);
    end;
  end;
  result:=_feld;
end;
end.

```

A.2 Quellcode vom Programm Adamlog (Grafikerzeugung)

```

program Adamlog;
uses
  Forms,
  Unit1 in 'Liu Ning\ADAM Projekt\readlogfile\Unit1.pas' {Form1};
{$R *.res}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, RMRLogfile, ExtCtrls, Meta, achsen, felder, contLines,
  plotXy, store, calc, Plots, xy_ext, zoom;
type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    Logfile1: TMenuItem;
    OpenFileDialog1: TOpenDialog;
    Plot1: TPlot;
    ZoomPopUpMenu1: TZoomPopUpMenu;
    procedure Logfile1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Plot1Paint(Sender: TObject);
  private
    { Private-Deklarationen }
    _logfile:TRMRLogfile;
    _afeld:TXyextended;
    function calcTemp(aFeld:TXyextended):Txyextended;
    function calcDaten(aFeld:TXyextended):Txyextended;

```

```

public
  { Public-Deklarationen }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Logfile1Click(Sender: TObject);
begin
  if opendialog1.Execute then
    _logfile.openfile(opendialog1.FileName);
  plot1.redraw;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
  _logfile:=TRMRLogfile.create(nil);
  _aFeld:=TXyextended.create(nil);
end;
function TForm1.calcDaten(aFeld:TXyextended):Txextended;
var i:integer;
begin
  result:=aFeld;
end;
function TForm1.calcTemp(aFeld:TXyextended):Txextended;
var i:integer;
const B=4540;
const RV=100E3;
const Ri=5E6;
const Uref=5;
var Rn,a: extended;
begin
  with aFeld do
    for i:=1 to N do
      begin
         $RN:=y[i]/(((Uref-y[i])/RV)-(y[i]/Ri));$ 
         $y[i]:=(B*298.15)/(B+\ln(RN/RV)*298.15)-298.15$ 
      end;
    result:=aFeld;
  end;
end;
procedure TForm1.Plot1Paint(Sender: TObject);
begin
  plot1.setField(calcDaten(_logfile.getDaten('SDR','p0','butr')), 'BUTRSPANNUNG');
  plot1.setField(calcDaten(_logfile.getDaten('SDR','p0','pzo')), 'PZOSPAUNUNG');
  plot1.setField(calcDaten(_logfile.getDaten('SDR','p0','fd1')), 'FOTODIODE1');
  plot1.setField(calcDaten(_logfile.getDaten('SDR','p0','fd2')), 'FOTODIODE2');
  plot1.setField(calcTemp(_logfile.getDaten('SDR','p0','tmp1')), 'LASERTEMP');
  plot1.setField(calcTemp(_logfile.getDaten('SDR','p0','tmp2')), 'RAUMTEMP');
  plot1.setField(calcTemp(_logfile.getDaten('SDR','p0','tmp3')), 'WASSERTEMP');
  plot1.setField(calcTemp(_logfile.getDaten('SDR','p0','tmp4')), 'EXTREMTEMP');
  plot1.execute('d:\user\adamprojekt.spz');
end;
end.

$DEFAULT VALUES
$VIEW 0 0 0.5 0.5
$RAHMEN D D D D
$BOX 0 0 24 5

```

```

$STYLE SOLID
$CLIP OFF
$UACHSE.mass_draw N
$SHRINK 0.75 0.75
$TEXT N C C 0.07 0.5 -0.12 Zeit [UT]
$SET_ANGLE 90
$TEXT N C C 0.07 -0.12 0.5 Spannung [V]
$REM COLOR WHITE
$COLOR RED
$PLOT LINE R BUTRSPANNUNG
$SET_ANGLE 0
$TEXTLINE N L C 0.05 0.1 1.1 Build up time reduction
$COLOR YELLOW
$PLOT LINE R PZOSPANNUNG
$SET_ANGLE 0
$TEXTLINE N L C 0.05 0.7 1.1 Piezo
$COLOR BLACK
$XACHSE D
$YACHSE D
$RACHSE D
$UACHSE D

```

```

$REM Plot 2
$VIEW 0.5 0.5 1 1
$RAHMEN D D D D
$BOX 0 18 24 40
$STYLE SOLID
$CLIP OFF
$UACHSE.mass_draw N
$SHRINK 0.75 0.75
$SET_ANGLE 0
$TEXT N C C 0.07 0.5 -0.12 Zeit [UT]
$SET_ANGLE 90
$TEXT N C C 0.07 -0.12 0.5 Temperatur [°C]
$REM COLOR BLACK
$COLOR AQUA
$PLOT LINE R LASERTEMP
$SET_ANGLE 0
$TEXTLINE N L C 0.05 0.1 1.1 Laserkopftemp
$COLOR ORANGE
$PLOT LINE R RAUMTEMP
$SET_ANGLE 0
$TEXTLINE N L C 0.05 0.6 1.1 Raumtemp
$COLOR BLACK
$XACHSE D
$YACHSE D
$RACHSE D
$UACHSE D

```

```

$VIEW 0 0.5 0.5 1
$RAHMEN D D D D
$BOX 0 -1.5 24 -0.5
$STYLE SOLID
$CLIP OFF
$UACHSE.mass_draw N
$SHRINK 0.75 0.75
$SET_ANGLE 0
$TEXT N C C 0.07 0.5 -0.12 Zeit [UT]

```

```

$SET_ANGLE 90
$TEXT N C C 0.07 -0.12 0.5 Spannung [V]
$REM COLOR WHITE
$COLOR BLUE
$PLOT LINE R FOTODIODE1
$SET_ANGLE 0
$TEXTLINE N L C 0.05 0.1 1.1 Seederlaser 532nm
$COLOR LIME
$PLOT LINE R FOTODIODE2
$SET_ANGLE 0
$TEXTLINE N L C 0.05 0.6 1.1 Seederlaser 1064nm
$COLOR BLACK
$XACHSE D
$YACHSE D
$RACHSE D
$UACHSE D

```

```

$VIEW 0.5 0 1 0.5
$RAHMEN D D D D
$BOX 0 5 24 25
$STYLE SOLID
$CLIP OFF
$UACHSE.mass_draw N
$SHRINK 0.75 0.75
$SET_ANGLE 0
$TEXT N C C 0.07 0.5 -0.12 Zeit [UT]
$SET_ANGLE 90
$TEXT N C C 0.07 -0.12 0.5 Temperatur [°C]
$REM COLOR BLACK
$COLOR FUCHSIA
$PLOT LINE R WASSERTEMP
$SET_ANGLE 0
$TEXTLINE N L C 0.05 0.1 1.1 Kühlwassertemp
$COLOR GRAY
$PLOT LINE R EXTREMTEMP
$COLOR BLACK
$XACHSE D
$YACHSE D
$RACHSE D
$UACHSE D

```

A.3 Quellcode der M-Files mit MATLAB

```

clear all;
close all;
clc;
[date,time,sdr,p0,tmp1,dt1,tmp2,dt2,tmp3,dt3,tmp4,dt4,butr,db,pzo,dp,fd1,df1,fd2,df2] =
textread('20110706.logfile.log','%s %s %s %s %s %f %s %f %s %f %s %f %s %f %s %f %s %f');

%%
fontsize = 15;
figure(1)
% subplot(2,1,1)
plot(datenum(time),dt1,'r');

```

```

datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, Zeit [UT]','FontSize',fontsize);
ylabel('Spannung [V]','FontSize',fontsize);

% subplot(2,1,2)
figure(2)
plot(denum(time),dt2,'.');
datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, time [UT]','FontSize',fontsize);
ylabel('Spannung [V]','FontSize',fontsize);

figure(3)
% subplot(2,1,1)
plot(denum(time),dt3,'.');
datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, Zeit [UT]','FontSize',fontsize);
ylabel('Spannung [V]','FontSize',fontsize);

figure(4)
plot(denum(time),dt4,'.');
datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, Zeit [UT]','FontSize',fontsize);
ylabel('Spannung [V]','FontSize',fontsize);

figure(5)
plot(denum(time),db,'.');
datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, Zeit [UT]','FontSize',fontsize);
ylabel('Spannung [V]','FontSize',fontsize);

figure(6)
plot(denum(time),dp,'.');
datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, Zeit [UT]','FontSize',fontsize);
ylabel('Spannung [V]','FontSize',fontsize);

figure(7)
plot(denum(time),df1,'.');
datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, Zeit [UT]','FontSize',fontsize);
ylabel('Spannung [V]','FontSize',fontsize);

```

```

figure(8)
plot(datenum(time),df2, '.');
datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, Zeit [UT]', 'FontSize',fontsize);
ylabel('Spannung [V]', 'FontSize',fontsize);
%%

figure(9)
hold on;
plot(datenum(time),dt1,'b. ');
plot(datenum(time),dt3,'m. ');
plot(datenum(time),dp,'y. ');
datetick('x','HH:MM','keplimits');
datetick('x','HH:MM','kepticks','keplimits');
%set(gca,'FontSize',fontsize);
xlabel('06-Jul-2011, Zeit [UT]');
ylabel('Spannung [V]');
legend('Laser-Kopf-Spannung','Kuehlwasser-Spann','Piezo-Spannung');
hold off;

figure(10)
[N,x]=hist(dt1,100);
bar(x,N*100/length(dt1));
xlabel('Spannung [V]');
ylabel('Haeufigkeit [%]');

figure(11)
[N,x]=hist(dt2,100);
bar(x,N*100/length(dt2));
xlabel('Spannung [V]');
ylabel('Haeufigkeit [%]');

figure(12)
[N,x]=hist(dt3,100);
bar(x,N*100/length(dt3));
xlabel('Spannung [V]');
ylabel('Haeufigkeit [%]');

figure(13)
[N,x]=hist(dt4,100);
bar(x,N*100/length(dt4));
xlabel('Spannung [V]');
ylabel('Haeufigkeit [%]');

figure(14)
[N,x]=hist(db,100);
bar(x,N*100/length(db));
xlabel('Spannung [V]');
ylabel('Haeufigkeit [%]');

figure(15)
[N,x]=hist(dp,100);
bar(x,N*100/length(dp));
xlabel('Spannung [V]');

```

```
ylabel('Haeufigkeit [%']);
```

```
figure(16)  
[N,x]=hist(df1,100);  
bar(x,N*100/length(df1));  
xlabel('Spannung [V]');  
ylabel('Haeufigkeit [%']);
```

```
figure(17)  
[N,x]=hist(df2,100);  
bar(x,N*100/length(df2));  
xlabel('Spannung [V]');  
ylabel('Haeufigkeit [%']);
```

Anhang B

Hydrostatische Grundgleichung

Mit Hilfe der Hydrostatischen Grundgleichung kann aus einem gemessenen Dichtprofile ein Temperaturprofil abgeleitet werden.

$$p_0 - p(z) = \int_{z_0}^z \rho(z) \cdot g(z) \cdot dz \quad (\text{B.1})$$

wobei ist **z** die Höhe und **P₀** der dazu gehörige Druck, **ρ** ist die Luftdicht und **g** ist die Erdbeschleunigung.

Abbildungsverzeichnis

2.1 Lidarprinzipskizze.....	4
2.2 Temperaturprofil zwischen 1 und 105 km am 5.10.2005.....	4
2.3 Termschema eines Nd:YAG-Lasers.....	5
2.4 Schematischer Aufbau des Nachweiszweiges des tageslichtfähigen RMR-Lidar.....	6
2.5 Q-Switch Build-up-time Reduction.....	7
2.6 Zusammenhänge zwischen erfassten Parametern.....	9
3.1 Strukturplan des in dieser Arbeit entwickelten Messsystems.....	10
3.2 Prinzipieller Aufbau des Transimpedanzverstärkers mit Fotodiode.....	11
3.3 Funktion der Fotodiode.....	13
3.4 Spektrale Empfindlichkeit der Hamamatsu S1223 Si PIN Fotodiode.....	14
3.5 Schematische Zeichnung der OPA380.....	15
3.6 Transimpedanzverstärker des OPA380 bei $C_{Diode}=10\text{ pF}$	16
3.7 Schaltung des Transimpedanzverstärkers.....	16
3.8 Grundprinzip eines Tiefpass-Filters.....	17
3.9a Simulationssoftware 1 Schritt.....	19
3.9b Simulationssoftware 2 Schritt.....	19
3.9c Simulationssoftware 3 Schritt.....	20
3.9d Simulationssoftware 4 Schritt.....	21
3.9e Toleranzauswahl der passiven Bauelemente.....	21
3.9f Vorgeschlagene Schaltung.....	21
3.10 Schaltung des aktiven Tiefpass-Filters.....	22
3.11 Frequenzgang des simulierten Tiefpass-Filters.....	22
3.12 Schaltung der Spannungsversorgung.....	23
3.13 Gesamtschaltplan für den Fotodiodenverstärker.....	25
3.14 PCB Layout und gelötete Platine für den Fotodiodenverstärker.....	25
3.15 Schaltung der Temperaturmessung.....	28

3.16 PCB Layout und gelötete Platine für die Temperaturmessung.....	29
3.17 Gehäuse-Vorderseite.....	30
3.18 Gehäuse-Rückseite.....	31
3.19 Gesamtplan der Laserüberwachung des Messsystems.....	33
3.20 Gehäuse: Draufsicht der aufgebauten Hardware.....	34
4.1 Programmablaufplan mit Delphi 6.....	36
4.2 Oberflächenkomponenten des ersten Programms.....	39
4.3 Struktur der gespeicherten Daten.....	39
4.4 Grafik der erfassten Parametern am 06.07.2011	40
5.1 Ablaufplan der grafischen Darstellung mit MATLAB	42
5.2 BUTR-Spannung sowie ihre Häufigkeit	44
5.3 Piezo-Spannung sowie ihre Häufigkeit	45
5.4 Fotodiode-Spannung 1 sowie ihre Häufigkeit.....	46
5.5 Fotodiode-Spannung 2 sowie ihre Häufigkeit.....	47
5.6 Laser-Kopftemperatur sowie ihre Häufigkeit.....	48
5.7 Raumtemperatur sowie ihre Häufigkeit	49
5.8 Kühlwassertemperatur sowie ihre Häufigkeit	50
5.9 Zusammenhänge zwischen verschiedenen Parametern.....	51

Tabellenverzeichnis

3.1 Übersicht drei Type der Tiefpass-Filters	18
3.2 alle Bauelemente der Platine für den Fotodiodenverstärker	26
3.3 alle Bauelemente der Platine für die Temperaturmessung	29
3.4 Beschreibung der Kanäle des ADAM4017 Moduls	32
3.5 Beschreibung aller Bauteile des Messsystems.....	33
4.1 Beschreibung einer 9 poligen RS-232 Schnittstelle.....	37

Literaturverzeichnis

Alpers M., Eixmann R., Fricke-Begemann C., et al.(2004). Temperature lidar measurements from 1 to 105 km altitude using resonance, Rayleigh, and Rotational Raman Scattering. *Atmo.Chem.phys.*, 4,793-800

Alpers M. et.al: The Rayleigh/Mie/Raman Lidar at IAP Kühlungsborn. *Journal of Aerosol Science*, 30 1999, 637–638

Gerding M., Höffner J., Kopp M., et al.(2010). Mesospheric temperature and aerosol soundings during day and night: Spectral and spatial filtering techniques. *In Reviewed and revised papers at the 25 th International Laser Radar Conference, St. Petersburg, Russia*, 67-70

Damaschke N.al.(2010). Lasermesstechnik. *Vorlesungsskript im Sommersemester 2010 an der Universität Rostock*

Holst H.al.(2007). Aufbau einer tageslichtfähigen Nachweisbank für ein Rayleigh-Lidar. *Diplomarbeit IAP Kühlungsborn Mai 2007*,21-30

Bauer, Helmbrecht (1991): Lasertechnik: Grundlagen und Anwendungen.

Gerding, M., Höffner, J., Raute,M., und F.-J- Lübken (2006), Observations of noctilucent clouds and temperature structure from 1-105 Km by co-located lidars at 54°N, in Proceedings of the SPIE symposium “*Lidar Technologies, Techniques, and Measurements for Atmospheric Remote Sensing II*”, Bd.6367, bearbeitet von U. Singh, SPIE, Bellingham, WA, doi:10.1117/12.689012.

Gerding, M, G. Baumgarten, G. von Cossart, R. Eixmann, J. Fiedler, J. Hildebrand, J. Höffner, T. Köpnick, M. Kopp, F.-J. Lübken, M. Priester (2009), Neue technische Entwicklungen bei den Lidarsystemen des IAP *Institutsbericht 2008/2009*,37-38

Datenblatt *Model 6500 Injection Seeder*, 3-4, Spectra Physics Injection Seeder

Jens Söder (2010). *Bachelor-Arbeit: Aufbau einer Wellenlängenstabilisierung für einen Nd:YAG-Laser,IAP*

Dai,Yongjiang (2010). Lidar-Technik. *Anwendung der Fotodiode in der Lidartechnik* Publishing House of Electronics Industry Beijing, 2010 334-341

Wikipedia: *Photodiode*. Online. zuletzt gelesen 04.2011.
<http://de.wikipedia.org/wiki/Photodiode>

Wikipedia: *Pin-Photodiode*. Online. zuletzt gelesen 04.2011.
<http://de.wikipedia.org/wiki/Pin-Diode>

Wikipedia: Signal-Rausch-Verhältnis. Online. zuletzt gelesen 04.2011.
<http://de.wikipedia.org/wiki/Signal-Rausch-Verh%C3%A4ltnis>

Datenblatt TI OPA380, Online. zuletzt gelesen 04.2011
<http://www.ti.com/lit/ds/symlink/opa380.pdf>

Datenblatt TI OPA137, Online. zuletzt gelesen 04.2011
<http://www.ti.com/lit/ds/symlink/opa137.pdf>

Thomas Kugelstadt, Chapter 16 Active Filter Design Techniques, *Literature Number SLOA088 von Texas Instruments*

Active Filter Design Application, Online. zuletzt gelesen 04.2011
<http://www.ti.com/tool/filterpro>

FilterPro™ User's Guide, *SBFA001C–June 1991–Revised February 2011, 1-37*

Ulrich Tietze, Christoph Schenk: *Halbleiter-Schaltungstechnik*. 12. Auflage. Springer, Berlin 2002, ISBN 3-540-42849-6.

ADAM 4000 Series, Data Acquisition Modules, User's Manual, *Edition 10.7 May 2008, Advantech Co., Ltd.*

Lerch, R.: *Elektrische Messtechnik*. Springer-Verlag. Berlin Heidelberg 1996.
ISBN 3-540-59373-X

Heinz Schmidt-Walter, *Aktive Filter, Elektronik*, Online, zuletzt gelesen 04.2011.
http://schmidt-walter.eit.h-da.de/el/skript_pdf/el_16.pdf

TRACO POWER DC/DC Converters TEL 2 Series, 2 Watt .Online. zuletzt gelesen 04.2011
<http://www.tracopower.com/products/tel2.pdf>

EAGLE Referenz-Handbuch ab 4.04, 3.Auflage, Copyright © 2001 CadSoft

Datenblatt AD586 by ANALOG DEVICES, Online. zuletzt gelesen 04.2011
http://www.analog.com/static/imported-files/data_sheets/AD586.pdf

NTC-Heißleiter, *Product Profile 2008 by EPCOS*, Online. zuletzt gelesen 04.2011
http://www.epcos.com/web/generator/Web/Sections/Publications/PDF/Hei_C3_9Fleiter,property=Data__en.pdf;Hei%C3%9Fleiter.pdf

Patrick Schnabel, NTC-Heißleiter, *Elektronik-Fibel*, 109-110

Pedrotti, F.; Pedrotti, L.; Bausch, W.; Schmidt, H.: *Optik für Ingenieure, Grundlagen*. 2. Auflage. Springer. Berlin 2002. ISBN 3-540-67379-2

Müller, R.: *Rauschen*. Zweite Auflage. Springer-Verlag. Berlin Heidelberg 1990.
ISBN 3-540-51145-8

RS-232, Online, zuletzt gelesen 08.2011.
<http://www.mikrocontroller.net/articles/RS-232>

Developer's Guide, Borland®Delphi™ 6 for Windows, 2001

Delphi-Hilfe, Online, zuletzt gelesen 07.2011, Delphi 6, 2001 Borland

Susanne Teschl (2001). *MATLAB Eine-Einführung*, Online zuletzt gelesen 09.2011.
<http://staff.technikum-wien.at/~teschl/MatlabSkriptum.pdf>

Danksagung

An dieser Stelle möchte ich den Personen danken, ohne die meine Masterarbeit nicht entstanden und nicht möglich gewesen wäre.

- Prof. Dr. Nils Damaschke für das Vertrauen in meiner Arbeitsweise, für seine Hilfe und Geduld.
- Prof. Dr. Franz-Josef Lübken für die Vergabe der Arbeit. Er ermöglichte es mir, meine Masterarbeit am IAP anzufertigen.
- Dr. Michael Gerding für die Betreuung dieser Arbeit, für die Hilfe und Geduld jederzeit.
- Herr Torsten Köpnick für die hilfreichen Anregungen und für die freundliche Unterstützung.
- Frau Maren Kopp und Herr Timo Viehl für Hilfe bei der Korrektur der Arbeit.

Ganz vielen Dank außerdem an:

Dr. Josef Höffner, Reik Ostermann, Michael Priester, Martin Schaeper, Dr. Martin Degner

Vielen Dank auch bei allen Mitarbeitern am IAP und am Institut für Allgemeine Elektrotechnik an der Universität Rostock.

Erklärung

Hiermit erkläre ich, die vorgelegte Arbeit selbständig und nur unter Zuhilfenahme der angegebenen Quellen und Hilfsmittel verfasst zu haben.

Rostock, den 20.9.2011

Liu, Ning