

Markus Paul Schöllauf

Matrikelnummer: 0330722

**Systemintegration und
Programmierung der Betriebssoftware
für den
Graz Mark1
Fluoreszenztomographen**

Diplomarbeit



Institut für Medizintechnik
Technische Universität Graz
Kronesgasse 5, A - 8010 Graz
Vorstand: Univ.-Prof.Dipl.-Ing.Dr.techn. Rudolf Stollberger

Betreuer und Begutachter:

a.o. Univ.-Prof. Dipl.-Ing. Dr. techn. Hermann Scharfetter

Graz, März 2012

*Ich danke meiner Freundin
Angelika für die
Unterstützung meines Studiums.*

*Ein Dankeschön gilt auch Manuel Freiberger
für seine tatkräftige Unterstützung bei der
Programmierung der Software*

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am 10. März 2012

Inhaltsverzeichnis

1 EINLEITUNG	9
1.1 FLUORESCENZTOMOGRAPHIE.....	9
1.2 REALISIERTES KONZEPT DER TU-GRAZ.....	9
1.3 AUFGABENSTELLUNG	10
1.3.1 Anforderungen an die Betriebssoftware	10
1.3.2 Anforderungen an die Kamera.....	11
1.3.3 Anforderungen an das Filterrad	11
1.3.4 Anforderungen an den Shutter	12
1.3.5 Anforderungen an die Hintergrundbeleuchtung.....	12
1.3.6 Anforderungen an die Sicherheitskomponenten.....	12
2 METHODEN	13
2.1 HARDWARE.....	13
2.1.1 Übersicht.....	13
2.1.2 Filterrad (Laser Abschwächung).....	17
2.1.3 Lasermodule	18
2.1.4 Shutter Komponente	20
2.1.5 Hintergrundbeleuchtung	21
2.1.6 Sicherheitskomponenten	21
2.1.7 Kamera mit Bildverstärker	24
2.1.8 Synthesizer (DDS)	24
2.1.9 Galvo-Spiegelsystem.....	25
2.1.10 Positionierung (Koordinatentisch).....	25
2.1.11 Endschalter für die Proben- und Laserarm Positionierung	26
2.1.12 Aufbau des USB-Moduls	27
2.2 SOFTWARE	27
2.2.1 Allgemeines	27
2.2.2 Kommunikationsmechanismus des USB-Moduls.....	30
2.2.3 Kamera Modul FTCamera	31
2.2.4 Positionierungsmodul FTPositioning	33
2.2.5 Laser Modul mit Filterrad	34
2.2.6 Galvo Modul.....	36
2.2.7 Shutter Modul	38
2.2.8 Synthesizer (DDS)	40
2.2.9 Backlight Modul	41
2.2.10 Synchrondemodulation	42
2.2.11 Image Viewer.....	43
2.2.12 Sicherheits-Watchdog.....	44
2.2.13 Bilddatencontainer FTImage.....	45
2.2.14 Script Engine	46

2.2.15 Konsole	48
2.2.16 Logger.....	48
2.2.17 Systemkonfiguration	49
2.3 RISIKOANALYSE	49
2.4 ABLEIT- UND BERÜHRUNGSSTROMMESSUNG	52
3 ERGEBNISSE UND DISKUSSION.....	55
3.1 RISIKOANALYSE	55
3.2 ABLEIT- UND BERÜHRUNGSSTROMMESSUNG	60
3.3 BETRIEBSSOFTWARE	61
3.4 HARDWARE.....	62
3.4.1 Allgemein	62
3.4.2 USB Kommunikation.....	63
4 ANHANG.....	64

Liste der verwendeten Abkürzungen

C++	Programmiersprache
CCD	charge coupled device
CCFL	<i>Cold cathode fluorescence lamp</i>
COM	<i>Serieller Kommunikationsport</i>
DDS	Direct digital synthesis
DNA	Desoxiribonukleinsäure
E_	Eintrittswahrscheinlichkeit
EEPROM	electrically eraseable programmable read only memory
F_	Folgen bei Fehlerfall
FISH	Fluorescence in situ hybridisation
FTI	<i>fluorescence tomography image</i>
Galvo	<i>Galvanometer</i>
GUI	<i>graphical user interface</i>
HF	Hochfrequenz
IDE	<i>Integrated development environment</i>
IEEE1394	Institute of Electrical and Electronics Engineers
INI	Initialisierungsdatei
KHz	Kiloherz
MCP	micro channel plate
MHz	<i>Megahertz</i>
NC	Normal Condition
OD	Optische Dichte
PWM	<i>Pulsweitenmodulation</i>
RS	Risikogruppe
RS232	Serielles Schnittstellenprotokoll
RSS	Restrisikogruppe
SFC	Single failure condition
SS	Sicherheitsstufe
SSH	<i>Secure shell</i>
SVN	<i>Subversion</i>
USB	Universal serial bus
XML	<i>extensible markup language</i>

Kurzfassung

Diese Diplomarbeit dokumentiert die Integration und Implementierung einer Betriebssoftware sowie einiger zusätzlicher Hardwarekomponenten für ein hardwaremäßig bereits teilweise existierendes Fluoreszenztomographiesystem. Die Hardware des Systems ist Teil diverser Diplom- bzw. Bachelor-Arbeiten von Studenten der Technischen Universität Graz. Im Rahmen einer interuniversitären Forschungsarbeit zwischen der Universität Aachen in Deutschland und der Technischen Universität Graz wird dieser Tomograph auf dem Gebiet der diffusen Fluoreszenztomographie Messung eingesetzt.

Die Implementierung der Betriebssoftware erfolgte in der Programmiersprache C++ unter Zuhilfenahme der Programmbibliothek *Qt* und wurde auf dem Betriebssystem Ubuntu 10.04, einer Linux-Distribution entwickelt. Es wurde jedoch bei der Entwicklung auf eine Möglichkeit der Portierung auf ein anderes Betriebssystem mit möglichst geringem Aufwand geachtet. Der softwaretechnische Entwicklungsansatz wurde so gewählt, dass sich – durch die Größe des Projektes bedingt – eine Modul-Bauweise der Hardware sowie softwareseitig ergab. Dadurch kann das System beliebig erweitert werden und neue Funktionen können ohne großen Aufwand in die Betriebssoftware integriert werden.

Die Betriebssoftware bietet eine zentrale Steuerungs-, Kommunikations- und Messplattform, die sämtliche Module bzw. Teile des Tomographie-Systems in ihrer Funktion und Bedienbarkeit erfasst und in einer intuitiv bedienbaren Schnittstelle vereint. Durch die Möglichkeit, Messprotokolle via Scripting auszuführen, wird ein mächtiges Instrument zur Erfüllung der geforderten Aufgaben bereitgestellt.

Schlüsselwörter:

Fluoreszenztomographie, Betriebssoftware, Laser,

Filterrad, Shutter

Abstract

In this diploma thesis the integration and implementation of an operating software for a fluorescence optical tomography system is described. Furthermore five hardware components were designed and implemented for this system. Some parts of the tomography system were already made within the scope of other diploma or bachelor thesis from students of Graz University of Technology. In a research cooperation between the Institute of Medical Engineering (TU Graz) and the University of Aachen (Germany) the fluorescence optical tomography system was used to perform various measurements in terms of fluorescence diffuse optical tomography.

The implementation of the operating software was done by the programming language C++ and the cross-platform application framework Qt. The development environment was set up under the operating system Ubuntu linux 10.04. Nevertheless, it has been taken care to be runnable also on other operating systems with little effort. The development approach of the software was selected in a way to get a modular hardware and software structure of the system. This allows it to be more flexible for future system expansion.

The operating software offers a collective control- communication- and measurement- platform which combines all modules and parts of the fluorescence tomography system for controlling and maintaining them. It provides an intuitive controllable graphical user interface for that purpose. Due to the ability to execute measurement protocols via a scripting language (JavaScript) the user gets a highly flexible solution for the desired intent.

Key words:

Fluorescence optical tomography, operating software, laser, attenuation wheel, shutter

1 Einleitung

1.1 Fluoreszenztomographie

Die Fluoreszenztomographie ist ein nichtinvasives Verfahren zur Rekonstruktion der 3D-Verteilung fluoreszierender Stoffe in Objekten. Die Einsatzgebiete der Fluoreszenztomographie sind vielfältig. Fluoreszierende Stoffe werden in der Medizin u.a. zur Markierung von Antikörpern sowie der Auswertung von DNA-Microarrays oder zur Diagnostik von Stoffwechselerkrankungen eingesetzt. Weitere Verwendung finden sie in der Krebsdiagnostik wo sie z.B. in der Fluoreszenz in situ Hybridisierung (FISH) zur Erforschung der Auswirkung bestimmter Substanzen im Gewebe Anwendung finden.

Ein weiteres Forschungsgebiet ist die Lebensdauerermessung von Fluorophoren. Die Lebensdauer (Lifetime) der Fluoreszenz von einem fluoreszierenden Farbstoff hängt u.a. von der Interaktion mit seinem Umfeld (Gewebe) ab. Dieser Sachverhalt kann dazu genutzt werden, um über eine Lebensdauerermessung auf chemische Zusammensetzung des Gewebes in unmittelbarer Umgebung des Fluorophors zu schließen. Da eine zeitaufgelöste Messung der Lebensdauer aufgrund der extrem kurzen Zeiten (im Nanosekundenbereich) sehr aufwendig ist, wird nach anderen Lösungen gesucht, um diese zu messen.

Eine Möglichkeit stellt die Messung der durch die zeitliche Verzögerung verursachten Phasenverschiebung eines mit Hochfrequenz modulierten Laserstrahles dar. Dieser Ansatz wurde auch in dem Fluoreszenztomographie System des Institutes für Medizintechnik der TU Graz aufgegriffen und implementiert.

1.2 Realisiertes Konzept der TU-Graz

Das System (Abbildung 1) soll Messungen an narkotisierten Mäusen, welchen Fluoreszenzfarbstoff verabreicht wurde, ermöglichen. Entwickelt und getestet wird jedoch mit zylindrischen Kunststoff-Phantomen mit eigener Farbstoffaufnahme. Die Probe wird auf einer Positioniereinheit fixiert. Diese Positionierung ermöglicht eine exakte Ausrichtung der Probe vor einer CCD-Kamera. Vor der Kamera befindet sich ein Bildverstärker (MCP) dessen Fotokathode mit einem hochfrequenten Signal

moduliert wird und somit einen extrem schnellen Verschlussmechanismus darstellt. Ein - mit der gleichen Hochfrequenz gespeister - Laser wird über eine Strahlableit-Einheit (Galvo-2-D-Scanner) auf die Probe gerichtet und regt den Fluoreszenzfarbstoff an. Das fluoreszierende Licht wird vom anregenden Licht durch ein Kantenfilter getrennt und von der Kamera erfasst. Durch die Modulation des Lichts und der MCP ergibt sich eine Synchrondemodulation, die dazu genutzt wird, die Phasenverschiebung des Fluorophors und somit die physiologischen Eigenschaften des umgebenden Gewebes zu rekonstruieren.

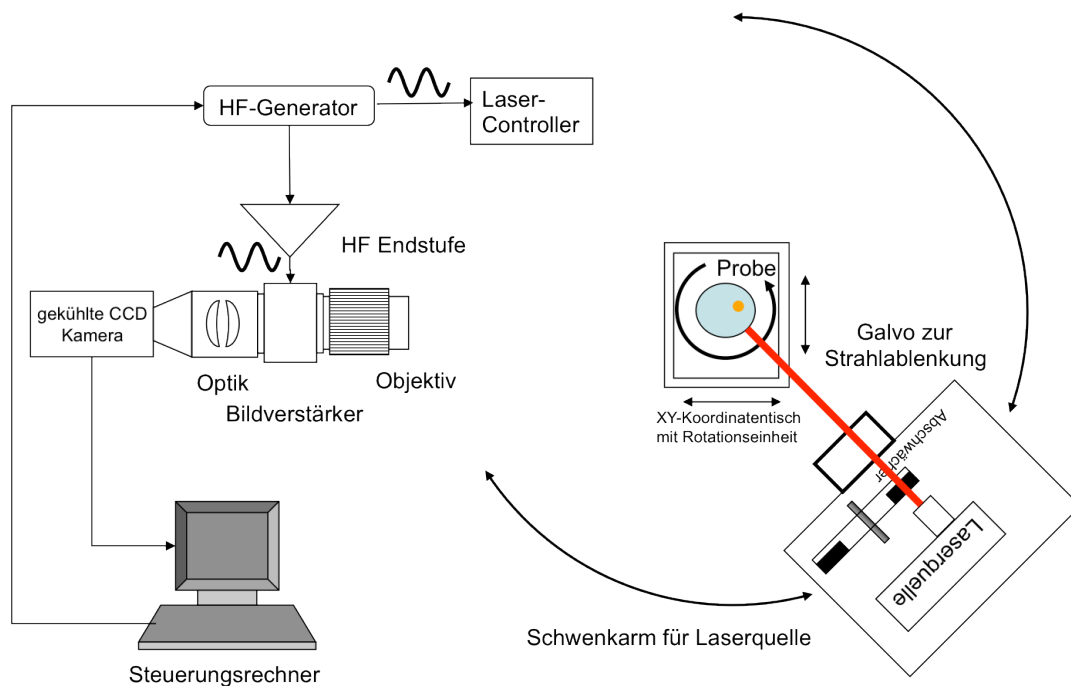


Abbildung 1: Übersichtsbild des Tomographen

1.3 Aufgabenstellung

1.3.1 Anforderungen an die Betriebssoftware

Durch die große Anzahl von Hardwaremodulen in diesem Fluoreszenztomographiesystem sowie die Notwendigkeit einer Ablaufsteuerung zur Ausführung einer Messung wird ein zentrales Instrument zur Steuerung dieses Systems sowie zur Entwicklung von Messabläufen bzw. Protokollen benötigt. Um diese Anforderungen zu erfüllen, soll eine **Betriebssoftware** entwickelt werden. Sie soll folgende Funktionalitäten abdecken:

- Zentrale Steuerung aller im System enthaltenen Komponenten
- Komfortables Entwickeln bzw. Verwalten von Messprotokollen
- Sicheres Ausführen von Messprotokollen sowie die Aufzeichnung des Ablaufs zur späteren Nachvollziehung
- Aufzeichnung und Bereitstellung der Messergebnisse durch ein möglichst einfaches Format
- Betrachtung und Analyse der aufgezeichneten Daten
- Möglichst einfache Erweiterbarkeit bzw. Portierbarkeit der Software

1.3.2 Anforderungen an die Kamera

Zur Zeit der Entwicklung der Betriebssoftware existierte kein Treiber für die Kamerahardware (ORCA II 4742-98-26LAG2, Hamamatsu) für das Betriebssystem Linux. Es soll eine Lösung gefunden werden, die Kamera unter Linux in vollem Funktionsumfang betreiben zu können, um eine Anbindung an den Linux-Steuerungsrechner zu ermöglichen.

1.3.3 Anforderungen an das Filterrad

Zur kontrollierten Abschwächung des Laserstrahls über mehrere Dekaden wird ein Filterrad benötigt. Zusätzlich soll eine Steuerung über die Betriebssoftware gewährleistet sein. Die Abschwächung soll in sechs Stufen (inklusive keine Abschwächung) der optischen Dichte (OD) erfolgen:

- | | |
|----------------------|-------|
| • keine Abschwächung | • OD2 |
| • OD 0.3 | • OD3 |
| • OD1 | • OD4 |

Die optische Dichte (OD) (oder Extinktion E_λ) ist wie folgt definiert:

$$E_\lambda = \log_{10} \left(\frac{I_0}{I} \right) \quad (1)$$

Dabei entspricht I_0 der Intensität des Lasers und I der abgeschwächten Intensität nach dem Filter. Wird z.B. die Abschwächung OD0.3 gewählt, wird die Laserstrahlung um den Faktor 2 abgeschwächt.

Weiters soll eine Möglichkeit zur manuellen Steuerung des Filtrerrades ohne Kommunikation mit dem Steuerungsrechner implementiert werden.

1.3.4 Anforderungen an den Shutter

Der Bildverstärker ist eine einstufige micro-channel-plate-Röhre (V7090U-71, Hamamatsu) [1], deren Fotokathode vor exzessiver Beleuchtung geschützt werden muss, um keinen Schaden zu nehmen. Zu diesem Zweck wird ein Schutzmechanismus in Form eines Shutters benötigt. Der Schutz wird durch eine zweifache Messung erreicht: Zunächst wird der Bildverstärker mit dem Shutter abgedeckt. Bei der darauffolgenden Belichtung des Objekts wird die Intensität des Streulichtes (sowie etwaiger Reflexe) am Shutter gemessen. Bleibt die Intensität unter einem zulässigen Wert, wird der Shutter geöffnet und das Objekt ein weiteres Mal belichtet. Der Bildverstärker kann nur während der zweiten Messung das Streulicht erfassen. Der Shutter soll über die Betriebssoftware bedienbar sein.

1.3.5 Anforderungen an die Hintergrundbeleuchtung

Zur exakten Bestimmung der Position und Geometrie der Probe sowie zur automatischen Belichtungseinstellung wird eine veränderbare Hintergrundbeleuchtung benötigt. Sie soll über die Betriebssoftware bedienbar sein.

1.3.6 Anforderungen an die Sicherheitskomponenten

Um einen in Hinblick auf Lasersicherheit unbedenklichen Betrieb des Fluoreszenztomographiesystems zu ermöglichen, werden diverse Schutzmechanismen wie z.B. Schlüsselschalter, Verriegelung und Türkontakt benötigt.

Eine gewichtige Rolle im Bezug auf Sicherheit spielt hier das Laser Modul. Es sollen alle softwaretechnischen Möglichkeiten genutzt werden, um einen sicheren Betrieb des Lasers zu ermöglichen.

Weiters soll eine Risikoanalyse sowie Ableitstrommessung des Fluoreszenztomographiesystems durchgeführt werden.

2 Methoden

2.1 Hardware

2.1.1 Übersicht

Der schematische Aufbau des Fluoreszenztomographie-Systems ist in Abbildung 2 ersichtlich und setzt sich aus folgenden Komponenten zusammen.

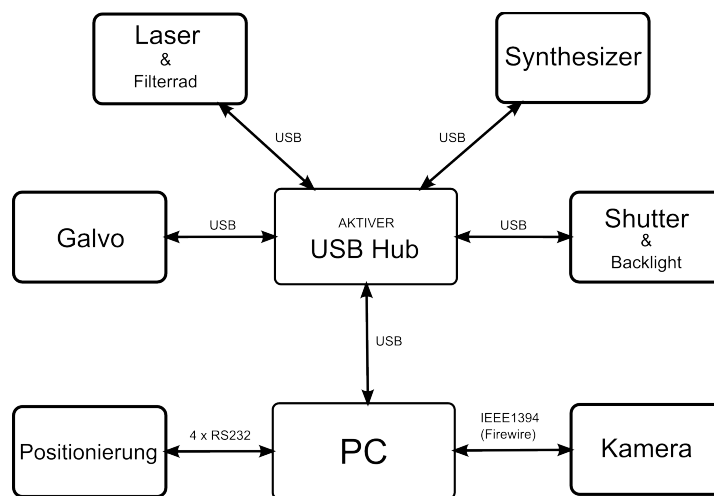


Abbildung 2: Schematischer Aufbau des Fluoreszenztomographie Systems

Kamera: Sie ist das ‚Herz‘ des Systems. Sie liefert die Bilddaten für die weitere Verarbeitung und wird per Firewire mittels dc1394 Bibliothek [2] an das System angebunden.

Bildverstärker: Er hat die Aufgabe, durch seine Funktion als Verschlussmechanismus eine optische Synchrondemodulation des zu erfassenden

Bildes durchzuführen. Dazu wird die Fotokathode mit einer, der Bias-Gleichspannung überlagerten, Hochfrequenzspannung (HF) betrieben.

Lasermodule: Der Laser bestrahlt die Probe mit moduliertem Laserlicht und regt diese zur Fluoreszenz an.

Filterrad: Dieses wird zur kontrollierten Abschwächung der Laserstrahlung benutzt. Es besteht aus 5 Filtergläsern die in den Strahlengang geschwenkt werden.

Synthesizer / DDS Board: Es dient zur Erzeugung der Hochfrequenz in einem Bereich zwischen 0 und 400 Mhz. Zusätzlich kann ein genau definierter Phasenversatz zwischen den beiden Signalausgängen erzeugt werden.

Positionierung: Ein Koordinatentisch wird zum genauen Positionieren der Probe in X und Y- Richtung sowie zur Rotation des Lasermodules um die Probe verwendet. Zum Einsatz kommt ein handelsüblicher Koordinatentisch, dessen Achsen mittels angeflanschter Schrittmotoren elektrisch angetrieben werden können.

Galvo-Spiegelsystem: Es besteht aus zwei beweglich gelagerte Spiegel samt Ansteuerung und ermöglicht eine präzise Positionierung des Laserstrahls.

Shutter: Er schützt den Bildverstärker vor übermäßiger Laserstrahlung bzw. zu hellem Umgebungslicht.

Hintergrundbeleuchtung: Sie dient zur Beleuchtung der Probe von hinten. Dadurch wird ein Schattenbild der Probe erzeugt, welches zur Rekonstruktion der konvexen Hülle sowie für diverse Abgleichmechanismen (z.B. „Autoscaling“) benutzt werden kann.

Sicherheitseinrichtung: Durch die im System vorhandenen Gefahrenquellen wie dem eingesetzten Laser der Klasse 3b oder die Hochspannungsversorgung für den Bildverstärker wurden diverse Sicherheitseinrichtungen notwendig. Diese bestehen aus folgenden Komponenten:

- Schlüsselschalter für den Laserbetrieb.
- Türkontakt zur automatischen Deaktivierung des Lasers bei Öffnung
- Mechanismus zur Verriegelung der Tomographen-Tür im Betrieb
- Warnleuchte am System Gehäuse zur Signalisierung von Laser-Strahlung

- Einhausung des Bildverstärkers und der Hochspannungsversorgung zur Vermeidung von Berührung spannungsführender Teile bzw. Abschirmung vor Hochfrequenz.
- Diverse Softwareimplementierungen von Sicherheitsmechanismen bez. Laseraktivität wie z.B. ein „Heartbeat“ Signal für das Lasermodul. Es stellt einen kontrollierten Zustand des Lasers bei Ausnahmesituationen der Betriebssoftware, wie z.B. Absturz sicher.

Aktiver USB-Hub: Durch die großen Leitungslängen sowie durch den erheblichen Stromverbrauch der Summe der Teilnehmer wird ein extern versorgter USB-Hub benutzt.

2.1.1.1 Übersicht über die Kommunikationsschnittstellen der Hardware

Die Kommunikation der Systemkomponenten mit dem Steuerungsrechner erfolgt mittels USB. Als grundsätzliche Modulhardware wird die Implementierung aus [3] benutzt. Sie wurde für alle Baugruppen eingesetzt, welche keine Standardschnittstelle ab Werk zur Verfügung stellen. Herzstück des Modules ist ein Mikrocontroller mit USB Client Profil von Microchip (PIC18F2550) [4], der mit einem Bootloader ausgestattet ist, mit dem der Firmware-Wechsel sehr bequem via USB durchgeführt werden kann. Das Systemdesign wurde so ausgelegt, dass jeder Baugruppe ein Software-Modul zugeordnet ist.

Es werden folgende Komponenten in je einem Hardware-Modul zusammengefasst:

Tabelle 1: Übersicht der benutzten Schnittstellen im System

Software-Modul	Funktionen	Schnittstelle
Camera	Kamera	IEEE1394 (Firewire)
Synthesizer	DDS-Board	USB
Shutter	Shutter und Hintergrundbeleuchtung	USB
Laser	Lasermodul und Filterrad	USB
Galvo	Galvo-System	USB
Positioning	Koordinatentisch mit C-Bogen	RS232 (4x)

Im Rahmen dieser Arbeit wurden neben der Programmierung der Betriebssoftware die folgenden Hardware-Module komplett bzw. in Kooperation mit anderen Studenten entwickelt und realisiert:

- Filtrerrad
- Lasermodul
- Shutter
- Hintergrundbeleuchtung
- Sicherheitskomponenten (Verriegelung, Türkontakt, Schlüsselschalter)

Weiters wurden folgende Komponenten verbaut, welche zugekauft wurden bzw. aus bestehenden Arbeiten von Studenten stammen:

- Kamera
- Synthesizer (DDS)
- Galvo-Scanner
- Positionierung

2.1.2 Filterrada (Laser Abschwächung)

Um die Intensität des Lasers kontrollieren zu können, wurde eine Abschwächer-Einheit entwickelt. Diese Einheit besteht aus einem Fächer aus Aluminium, auf dem fünf 1/2" ND-Filtere (Typ NE5xxB, Thorlabs) [5] in verschiedenen optischen Dichten (OD) angeordnet sind.

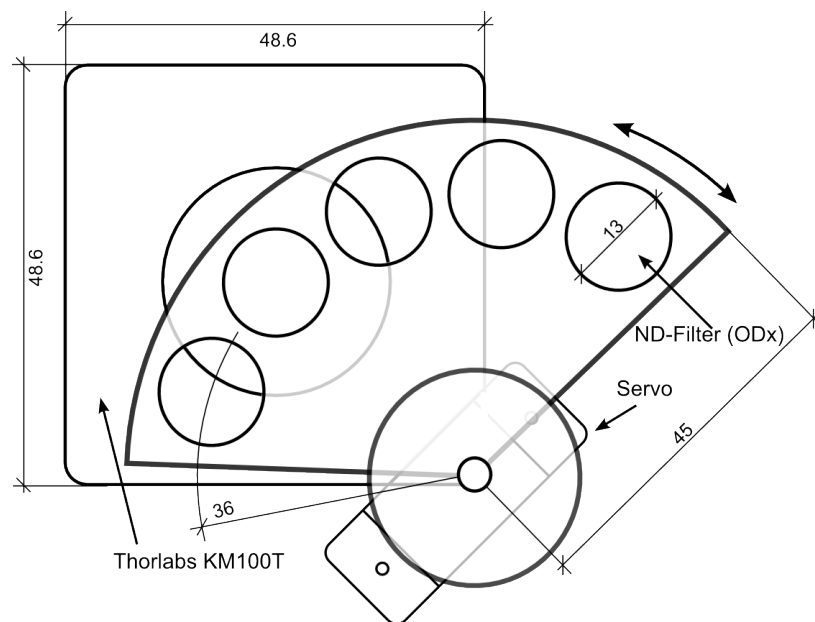


Abbildung 3: Aufbau der Abschwächer Einheit

Dieser Fächer ist auf der Achse eines analogen Miniatur-Modellbauservos [6] angebracht. Das Servo ist an einem Optikhalter des Typs KM100T [7] montiert (Abbildung 3). Durch Rotation des Servos kann nun der Fächer in den Strahlengang des Lasers geschoben und entsprechend der Positionierung des Fächers eine Abschwächung des Laserstrahls erzielt werden. Die Filterscheiben wurden mithilfe von Objektträger-Glas als Planaufgabe in die Öffnungen eingeklebt. Dabei wurden die optischen Dichten OD 0.3, OD1, OD2, OD3 und OD4 benutzt. Wird keine Abschwächung benötigt, wird der Fächer (Filterrad) komplett aus dem Strahlengang herausgeschoben.

Zur Steuerung des Servos kommt das USB-Modul [3] zum Einsatz. Die hierzu entwickelte Firmware verwendet die Capture-Compare-Funktion des Controllers, um eine stabile und auf effektive 11 Bit aufgelöste Positionierung mittels Servo-Protokoll zu erzeugen. Dadurch ist eine exakte Einstellung der Position der Filter im Strahlengang möglich. Nach Speicherung der sechs Positionen (fünf ODs und einmal ohne Abschwächung) im EEPROM des Controllers kann durch zwei, auf einer

Sandwich-Platine über dem USB-Modul angebrachte Taster eine schrittweise Laser-Abschwächung auch manuell ohne Kommunikation mit dem Host durchgeführt werden. Die Justierung der Filter-Positionen erfolgt mithilfe des Laser-Steuerungsmoduls im Hauptprogramm *ftcontrol*. Nähere Informationen finden sich im Bedienungshandbuch im Anhang. Das benutzte USB-Modul aktiviert bzw. deaktiviert auch den Laser.

2.1.3 Lasermodul

An dieses Modul wurden vor allem sicherheitstechnische Anforderungen bei Ein- und Ausschalten des Lasers gestellt.

Die Hardware wurde in einer Kooperation mit Timucin Karaca realisiert, der es als Teil seiner Bachelorarbeit [8] behandelt. Die Laserdiode des Typs ADL-66701HL (Laser Components Deutschland) [9] besitzt eine Leistung von max. 100mW und eine Wellenlänge von 658nm (rot). Es handelt sich um einen Laser der Klasse 3b.

Da der Laser mit etwa 100 MHz amplitudenmoduliert wird musste Sorge getragen werden, dass zu keinem Zeitpunkt hochfrequente Energie **ohne** Bias-Versorgung die Laserdiode erreichen kann. Ansonsten wäre das Risiko zu hoch, durch die negativen Halbwellen des HF-Signals den Laser zu zerstören.

Aus diesem Grund wurde eine Vorrangschaltung (Abbildung 4) entwickelt, welche bei Einschalten der Versorgung zuerst die Bias-Versorgung, dann die HF aufschaltet. Bei Abschalten wird zuerst die HF abgekoppelt und dann die Bias-Versorgung. Die Ab- und Ankoppelung wird mit einem HF-Relais durchgeführt. Nähere Informationen finden sich in [8].

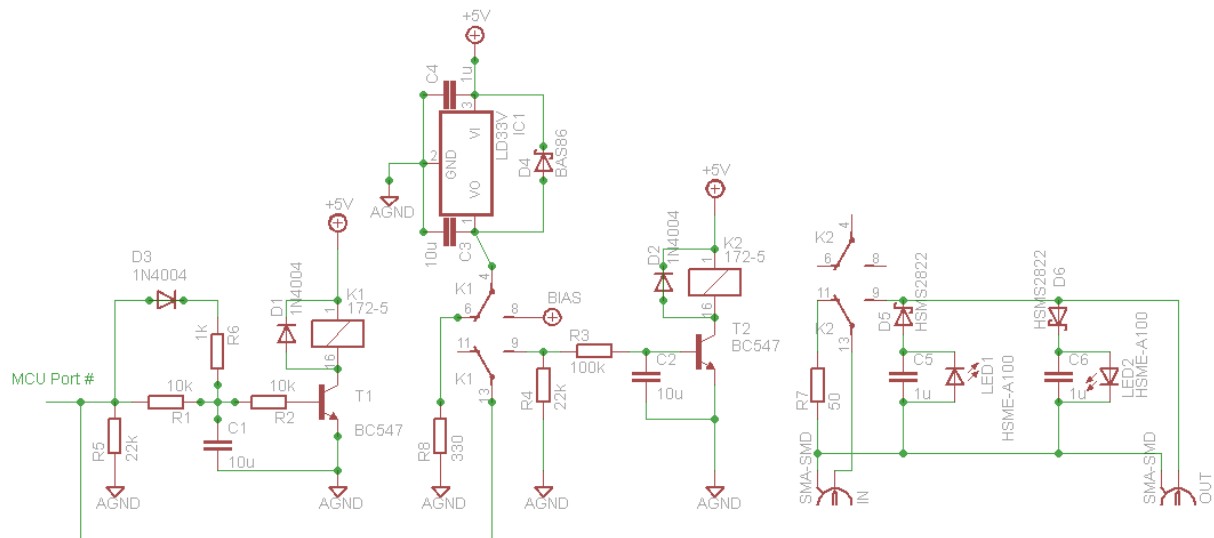


Abbildung 4: Versorgung und Vorrangschaltung für den Laser

Tabelle 2 zeigt die Ein- bzw. Ausschaltzeiten der Bias-Versorgung und der HF nach dem Aktivieren bzw. Deaktivieren durch den Controller.

Tabelle 2: Ein-Ausschaltzeiten von Bias und HF - Einschalten: uC Pin wechselt von LOW auf HIGH, t=0 - Ausschalten: uC Pin wechselt von HIGH auf LOW, t=0

	Einschalten	Ausschalten
Bias	5 ms	140 ms
HF	35 ms	50 ms

Zusätzlich wurde eine Möglichkeit geschaffen, bei Öffnen der Tomographen-Tür den Laser zu deaktivieren, um so unbeabsichtigtes Austreten von Laserstrahlung in den Raum zu verhindern. Nähere Information finden sich in der Spezifikation der Sicherheitskomponenten in Kapitel 2.1.6.

2.1.4 Shutter Komponente

Der Shutter besteht aus einem Arm aus dünnem Messingblech, der die Größe des Objektivs des Bildverstärkers besitzt und vor diesem positioniert ist (Abbildung 5). Dieser Arm ist an der Achse eines analogen Modellbau-Servos [10] angebracht. Dadurch ist es möglich, den Arm entweder aus dem Bildbereich des Objektivs zu drehen und es damit freizugeben, oder das Objektiv gänzlich abzudecken.

Weiters ist eine Solarzelle (Polykristallin, Conrad) [11] auf dem Arm angebracht, welche bei abgedecktem Objektiv das Umgebungslicht vor diesem erfasst. Die generierte Spannung stellt ein Maß für die potentielle Exposition des Bildverstärkers durch das Umgebungslicht dar.

So wird verhindert, dass es bei ungünstiger Laser Positionierung bzw. durch Streulicht zur Übersteuerung des Bildverstärkers und einer eventuellen Schädigung der Fotokathode kommt.

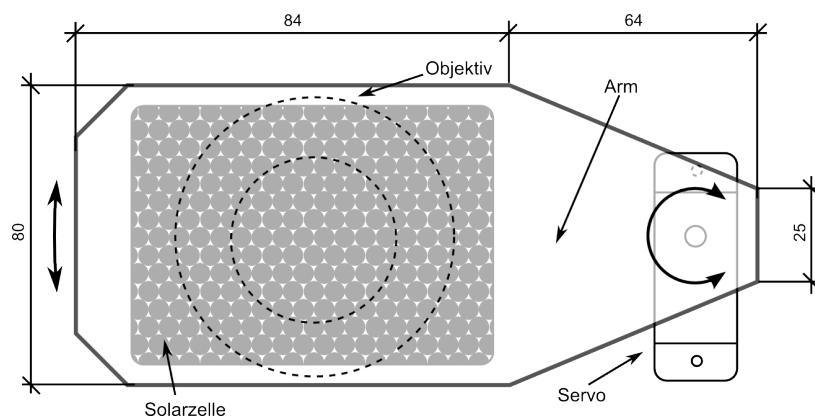


Abbildung 5: Aufbau des Shutter

Zur Steuerung des Servos kommt ein weiteres USB-Modul zum Einsatz. Die Firmware hierzu wurde von der Laser Komponente übernommen und entsprechend abgeändert. Hinzu kam die softwareseitige Signalaufbereitung des ‚excessive light‘-Eingangs von der Steuerplatine des Shutters. Sie führt eine Impulsformung durch und stellt so sicher, dass auch ein kurzzeitiges ‚Darüberstreichen‘ des Lasers über die Apertur des Objektivs erkannt wird. Die Justierung der Positionen für Shutter ‚geöffnet‘ und ‚geschlossen‘ erfolgt mithilfe des Shutter-Steuermoduls im Hauptprogramm *ftcontrol*. Nähere Informationen finden sich in der Bedienungsanleitung im Anhang. Das benutzte USB-Modul verwaltet und steuert auch die Hintergrundbeleuchtung.

Die ursprüngliche Shutter-Realisierung wurde im Rahmen der Projektarbeit von Gerald Schwarz [12] entwickelt. Sie verwendete ein ausgedientes CD-Laufwerk, dessen Tray als Shutter diente. Die Ansteuerung des Tray-Motors (nicht mehr verwendet), die Aufbereitung der Solarzellen-Spannung sowie die PWM-Endstufe für die Hintergrundbeleuchtung sind Teil jener Arbeit.

2.1.5 Hintergrundbeleuchtung

Die Beleuchtungseinheit besteht aus einer Hintergrundbeleuchtung eines alten Laptopdisplays, die hinter der Probe (an der Innenwand des Tomographen) in einem Holzkasten mit entsprechender Öffnung montiert ist. Das Leuchtmittel ist eine Kaltkathodenröhre (CCFL). Zur Reduzierung der maximalen Lichtintensität der Röhre wird ein Abschwächer-Filter davor eingesetzt.

Die Ansteuerung der Beleuchtung erfolgt über das USB-Modul des Shutters. Es wird eine Pulsweitenmodulation (PWM) mit einer Arbeitsfrequenz von ca. 12 kHz benutzt. Das PWM-Signal steuert eine Endstufe, bestehend aus einem Leistungstransistor in Emitterschaltung und einem RC-Glied. Da die CCFL eine Spannung von ca. 500V – 600V für die Zündung benötigt, wird sie über ein 12V-CCFL Inverter Modul (Conrad ArtNr. 720256) [13] angesteuert. Dieses Inverter-Modul wird normalerweise mit einer Spannung von 12V betrieben, ermöglicht aber durch Verringerung dieser Spannung eine Helligkeitssteuerung der CCFL in gewissen Grenzen. Das Inverter-Modul wird von der Endstufe versorgt. Damit wird eine softwareseitige Helligkeitssteuerung des CCFL erreicht. Die prinzipbedingten Eigenschaften (wie z.B. Zündspannung) der CCFL werden per Software behandelt.

2.1.6 Sicherheitskomponenten

2.1.6.1 Türverriegelung

Zweck der Türverriegelung ist es, vor einer versehentlichen Exposition einer Laserstrahlung während des Betriebs des Tomographie-Systems zu schützen, indem die Zugangstür (Holz-Klapptür) verriegelt wird. Weiters wird vermieden, dass ein

unbeabsichtigtes Öffnen der Tür zu viel Umgebungslicht auf den Bildverstärker fallen lässt und ihn dadurch beschädigt. Die Verriegelung besteht aus einem Aluminiumbügel in Hakenform, der drehbar und etwas entfernt von seinem Schwerpunkt gelagert ist (Abbildung 6). So befindet er sich im Ruhezustand immer in ‚geöffneter‘ Stellung. Durch einen Hubmagneten (TDS-03A, Conrad ArtNr: 503538) [14] kann diese Ruheposition geändert werden, indem bei Anziehen des Magneten der Bügel in die ‚geschlossene‘ Stellung kippt und mit seinem Haken in eine an der Tür angebrachte Klinke eingreift. So wird die Tür bei Bedarf verriegelt und kann nicht mehr geöffnet werden.

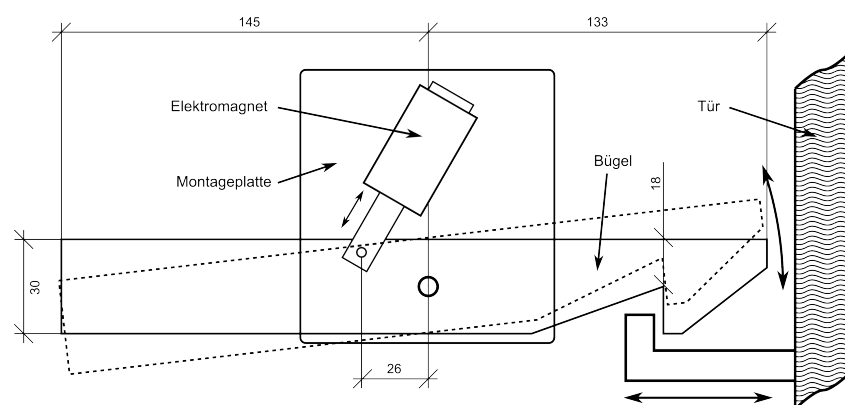


Abbildung 6: Aufbau des Verriegelungsmechanismus der Tomographen-Tür

Die Verriegelung ist aktiv, sobald das Netzteil des Lasermoduls eingeschaltet wird. Zur Entriegelung der Tür wird der Schlüsselschalter für den Einrichte- bzw. Wartungsbetrieb benutzt. Nähere Informationen zum Schlüsselschalter finden sich in Kapitel 2.1.6.3.

2.1.6.2 Türkontakt

Um sicherzustellen, dass nur dann Laserstrahlung erzeugt wird wenn die Tür des Tomographen geschlossen ist, ist ein Türkontakt verbaut. Er überwacht die Stellung der Tür und gibt die Ansteuerung des Lasers nur dann frei, wenn die Tür geschlossen ist *oder* der Schlüsselschalter auf Einrichte- bzw. Wartungsbetrieb eingestellt ist.

2.1.6.3 Schlüsselschalter

Zur Einrichtung und Justierung des Systems bzw. zur Vorbereitung von speziellen Messungen ist es erforderlich, Laserstrahlung bei geöffneter Tomographen-Tür bereitzustellen. Hierzu existiert ein Schlüsselschalter [15] an der rechten Seite des Tomographie-Systems. Er hat zwei Schaltpositionen: **Normalbetrieb** und **Wartungsbetrieb**.

Normalbetrieb: Dieser Modus stellt den **normalen** Betriebszustand des Systems dar. In diesem Modus werden sämtliche Messprotokolle abgearbeitet. Er bietet alle Sicherheitsaspekte und Mechanismen, die für einen sicheren Betrieb des Tomographie-Systems erforderlich sind. Nähere Informationen zu sicherheitstechnischen Fragestellungen finden sich in Kapitel 2.3.

Wartungsbetrieb: Dieser Modus ist **nur für temporäre Einrichte-, Wartungs- oder Reparaturarbeiten** vorgesehen und geeignet. Er setzt folgende Sicherheitsvorkehrungen außer Kraft:

- Die Sicherheitsverriegelung der Tomographen-Tür wird deaktiviert
- Die Funktion des Türkontakts zur Abschaltung der Laserstrahlung beim Öffnen der Tür (2.1.6.2) wird überbrückt.

Zusätzlich wird eine Warnlampe an der rechten Seite des Tomographen aktiviert, um das erhöhte Sicherheitsrisiko optisch anzuzeigen (vgl. Abbildung 7).

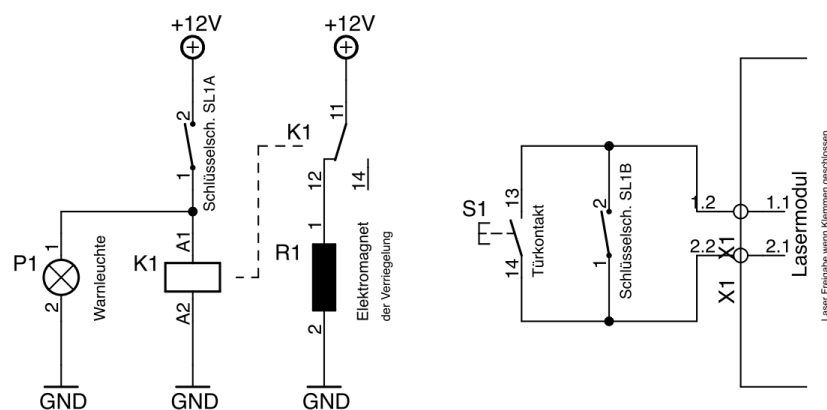


Abbildung 7: Sicherheitsschaltung für die Laser Freigabe und Türverriegelung

2.1.7 Kamera mit Bildverstärker

Zur Erfassung der Bilder im Messprotokoll wird die mit Peltier-Elementen gekühlte Kamera ORCA II 4742-98-26LAG2 (Hamamatsu) [16] verwendet. Sie besitzt eine Bildgröße von 512x512 Pixel und eine Auflösung von maximal 16 Bit. Es gibt zwei Video-Modi:

- High-Speed Readout (12 Bit Auflösung)
- High-Precision Readout (16 Bit Auflösung)

Die Kamera ist DCAM (I IDC) zertifiziert und kann dadurch über eine offene dc1394 Bibliothek [17] angesteuert werden.

Direkt vor der Kamera sitzt der MCP-Bildverstärker (V7090U, Hamamatsu) [1]. Er benötigt eine Nachbeschleunigungsspannung von ca. 6 kV und ist sehr lichtempfindlich. Durch Modulation der Spannung an seiner Fotokathode kann ein Verschlussmechanismus (Shutter) realisiert werden, der bei der Erfassung der Phasenbilder zur Synchrondemodulation des Streu- bzw. Fluoreszenzlichtes benutzt wird, welches vom modulierten Laser angeregt wird. Details zur Synchrondemodulation finden sich in Kapitel 2.2.10.

2.1.8 Synthesizer (DDS)

Die Erzeugung der für die Synchrondemodulation benötigten Hochfrequenz sowie die kontrollierte Phasenverschiebung zwischen zwei HF-Quellen übernimmt ein Direct Digital Synthesis (DDS) Modul. Es kann Frequenzen von einigen kHz bis zu ca. 400 MHz generieren und besteht grundsätzlich aus zwei Komponenten:

- Takterzeugung (AD9520)
- 2x Direct Digital Synthesizer Chip (AD9912)

Beide DDS-Chips besitzen zu Beginn einen bestimmten Phasenversatz zueinander. Aufgrund der gemeinsamen Taktquelle ändert sich dieser Versatz nicht. Weiters ist es möglich, einen definierten Winkel zwischen den beiden DDS-Kanälen einzustellen. Alle Komponenten am Modul werden über einen Mikrokontroller (PIC18F2550, Microchip Technology Inc.) [4] gesteuert. Zusätzlich besitzt dieser

Kontroller eine USB-Schnittstelle, welche für die Kommunikation zum Steuerungsrechner benutzt wird. Er enthält den gleichen USB-Bootloader wie die USB-Module [3]. Detaillierte Informationen über das DDS-Modul finden sich in der Projektarbeit [18] für dieses Modul.

2.1.9 Galvo-Spiegelsystem

Das Spiegelsystem besteht aus zwei Galvo-Motoren (Thorlabs), auf deren Achsen sich Spiegel befinden. Sie werden so montiert, dass ein auf sie treffender Lichtstrahl in zwei Richtungen abgelenkt werden kann (Azimut und Elevation). Dadurch wird eine genaue Positionierung des Laserstrahls auf der Probe ermöglicht.

Auch hier übernimmt die Steuerung der Galvos ein Mikrokontroller (PIC18F2550, Microchip) [4] mit der USB-Bootloader Firmware und einer USB-Schnittstelle zum Steuerungsrechner. Detaillierte Informationen über das Spiegelsystem finden sich in der Bachelorarbeit von Richard Zettl [19].

2.1.10 Positionierung (Koordinatentisch)

Der Koordinatentisch samt angeflanschter Schrittmotoren wurde im Rahmen der Diplomarbeit von Martin Toferer [20] entwickelt. Jedoch wurde die Ansteuerung (Labview [21]) und Kommunikation zu den Motoren komplett überarbeitet. Sie ist Hauptbestandteil einer weiteren Projektarbeit [22].

Die Positionierung besteht aus einem XY-Koordinatentisch, bei dem beide Achsen mithilfe von angeflanschten Schrittmotoren elektronisch verstellbar sind. Dabei werden Motoren des Typs *PANdrive* (Trinamic) [23] eingesetzt.

Weiters ist am oberen Teil des Tisches ein C-Bogen aus Metall angebracht. An diesem ist ein drehbar gelagerter Laser-Arm montiert, der durch einen Schrittmotor angetrieben wird. Zusätzlich befindet sich in der Rotationsachse des Laser-Arms ein weiterer Schrittmotor, der die Rotation der Probe übernimmt.

Jeder Motor besitzt seine eigene Steuerungs- und Endstufen-Elektronik, welche an der Hinterseite der Motoren (Sandwich-Bauweise) sitzt. Die Kommunikation erfolgt

mittels RS232-Protokoll. Zu diesem Zweck wurden am Steuerungsrechner zwei Dual-COM-Port-Karten installiert.

2.1.11 Endschalter für die Proben- und Laserarm Positionierung

Zur Referenzierung der Achse für die Probenrotation wird ein unipolarer Hallschalter (H501, PIC) [24] mit fest eingestellter Schaltschwelle benutzt. Er aktiviert seinen Open-Drain-Ausgang bei Annäherung eines hinreichend starken magnetischen Südpoles. Er ist am unteren Schenkel des C-Bogens mithilfe einer Metallvorrichtung befestigt (vgl. Abbildung 8). Ein Magnet ist seitlich am Probenteller so aufgeklebt, dass er bei Rotation des Tellers den Hallschalter aktiviert und so den Referenzwinkel für die Probenrotation festlegt.

Der Schrittmotor (PANdrive, Trinamic) [23] bietet die Möglichkeit über Eingänge für End- und Referenzschalter eine automatische Referenzfahrt durchzuführen. Diese Möglichkeit wurde durch Anschließen des Hallschalters an den Referenzschalter-Eingang genutzt. Zur Durchführung einer Referenzfahrt wird lediglich ein entsprechendes Kommando per RS232 gesendet.

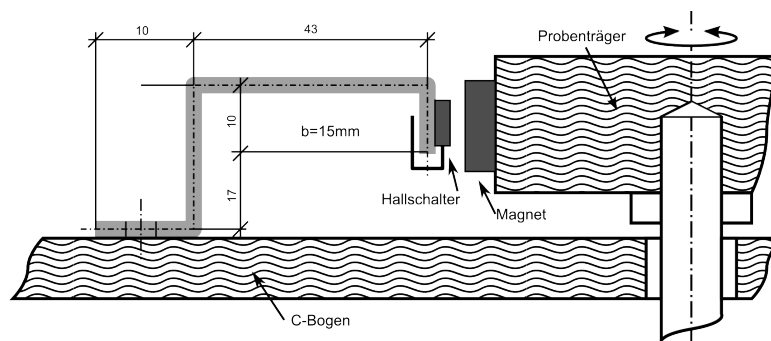


Abbildung 8: Aufbau des Endschalters für die Probenrotation

Der Endschalter für den Laserarm ist oben am C-Bogen neben dem Schrittmotor befestigt. Dabei ist er so montiert, dass ein am Laserarm aufgeklebter Magnet kurz vor dem mechanischen Anschlag des Arms eine Aktivierung auslöst und so den Referenzwinkel definiert.

2.1.12 Aufbau des USB-Moduls

Die meisten Hardwarekomponenten des Fluoreszenztomographiesystems benutzen für die Kommunikation mit dem Steuerungsrechner das USB-Modul von Manuel Freiberger [3]. Herzstück des Moduls bildet der Mikrokontroller PIC18F2550 (Microchip) [4]. Er beinhaltet einen Bootloader, der für ein komfortables Aufspielen der Firmware via USB entwickelt wurde. Hierzu befindet sich das Kommunikationsprogramm *bootloader* am Steuerungsrechner im Ordner `ftcontrol/bootloader`. Das USB-Modul beinhaltet weiters einen Quarz für den Systemtakt sowie Standardkomponenten für die Beschaltung des Reset-Eingangs für den Controller. Sämtliche Pins des Kontrollers sind auf Pfostensteckerleisten gelegt, um das Modul möglichst flexibel und universell verwenden zu können.

Softwaretechnisch baut die Kommunikation auf der Open-Source Programmbibliothek *libusb* [25] zum Schreiben und Lesen von USB-Geräten auf. Kommandos an das Modul werden im *Control-Transfer-Modus* gesendet. Die detaillierten Steuerungsbefehle werden im Kapitel 2.2 für die entsprechenden Module behandelt.

2.2 Software

2.2.1 Allgemeines

Als integrierte Entwicklungsumgebung (IDE) für die Betriebssoftware wurde Qt [26] in der Version 4.7 unter dem Betriebssystem Ubuntu Linux 10.04 [27] gewählt.

Qt ist eine C++-Klassenbibliothek für die plattformübergreifende Programmierung grafischer Benutzeroberflächen. Neben der Entwicklung grafischer Benutzeroberflächen bietet Qt umfangreiche Funktionen zur Internationalisierung sowie Datenbankfunktionen und XML-Unterstützung an und ist für verschiedene Betriebssysteme bzw. Grafikplattformen, wie X11 (Unix-Derivate), Mac OS X, Windows und als PDA-Version erhältlich. Qt wird insbesondere in den Bibliotheken der KDE Software Compilation 4 verwendet, welche gleichzeitig das prominenteste Vorzeigebeispiel der Klassenbibliothek darstellt. [28]

Das gesamte Hauptprogramm *ftcontrol* sowie die Firmware der Hardware-Module und Dokumentation steht unter Versionskontrolle (SVN) [29]. Das Repository liegt auf einem Repository-Server der TU-Graz und kann wie folgt eingesehen bzw. ausgecheckt werden:

```
svn co svn+ssh://svn@129.27.144.113/srv/svn/ftcontrol
```

Um einen Zugang zum Repository zu erhalten, ist der SSH-Public Key am Repository-Server einzutragen und freizugeben. Dies wird vom Administrator des Institutes für Medizintechnik der TU Graz durchgeführt.

Weiters wurde eine detaillierte softwaretechnische Dokumentation des Steuerungsprogramms *ftcontrol* mithilfe von DoxyGen [30] eingerichtet und generiert. Diese Dokumentation findet sich unter

```
ftcontrol/trunk/doc/html/index.html
```

und liegt im HTML-Format vor.

Das Projekt hat folgende Ordnerstruktur:

bootloader	doc	usbdiagnose
camera	galvo	watchdog
common	laser	
control	phaseshifter	
david	positioning	
dds	shutter	

bootloader: Hier befinden sich die Quellen bzw. das Binary des Bootloaders für die USB-Module.

david: Der Oberflächen-Scanner DAVID [31] ist ein Studentenprojekt und wurde für den Tomographen entwickelt, um eine 3D-Oberflächenabbildung der Probe zu erzeugen. Es benutzt u.a. die Objektrotation der Positioniereinrichtung, um verschiedene Erfassungswinkel der Probe einzustellen. Um die weitere Verwendung des Scanners zu gewährleisten, wurde eine Möglichkeit geschaffen, die Steuerung der Objektrotation der Positioniereinrichtung bei Bedarf an den Scanner zu übergeben. Weiters sollte kein Eingriff an der Scannersoftware erfolgen müssen. So

wurde eine COM-Port-Weiterleitung von einem freien Port auf den Port der Objektrotation eingerichtet.

camera: Quellen für das Softwaremodul **Camera**.

common: Quellen für die Low-Level-Kommunikation mit den USB-Modulen

control: Quellen des Hauptprogramms *ftcontrol*

dds: Quellen für das Softwaremodul **Synthesizer (DDS)**

doc: Ordner für die DoxyGen-Dokumentation

galvo: Quellen für das Softwaremodul **Galvo**

laser: Quellen für das Softwaremodul **Laser**

phaseshifter: Quellen für eine erste Version des *Phaseshifter* Softwaremoduls mit älterer Hardware (wird nicht mehr produktiv verwendet)

positioning: Quellen für das Softwaremodul **Positioning**

shutter: Quellen für das Softwaremodul **Shutter**

usbdiagnose: Quellen für ein USB-Diagnosetool zur Wartungs- bzw. Fehlersuche

watchdog: Quellen für den Sicherheits-Watchdog des Hauptprogramms

Im Folgenden werden die grundlegende Struktur und technische Besonderheiten der jeweiligen Softwaremodule behandelt. Darüber hinaus werden die Kommunikationsparameter der Module erläutert. Detaillierte Informationen wie Implementierungen oder Deklarationen finden sich in der *DoxyGen*-Dokumentation. Weiters steht der gesamte Quellcode zur Verfügung.

Alle Software-Module besitzen eine entsprechende GUI, die über das Hauptmenü des Programms *ftcontrol* zugänglich sind. Außerdem können die meisten Module auch in einem „stand-alone“-Betrieb ausgeführt werden ohne das Hauptprogramm starten zu müssen.

2.2.1.1 Signal-Slot Mechanismus von Qt

Das Event-Handling von Qt benutzt einen Mechanismus der es erlaubt, durch eine Art „Abonnieren von Informationsquellen“ Benachrichtigungen zu erhalten, wenn sich die zugrundeliegende Information ändert. Diese Informationsquellen werden *Signale* genannt und mit dem Zugriffsmodifikator *signals* deklariert. Möchte z.B. eine Klasse A Änderungen des Betriebszustands der Klasse B mitgeteilt bekommen, wird in der Klasse A eine Methode implementiert, die dann ausgeführt wird, wenn eine Änderung in der Klasse B auftritt. Diese Methode wird *Slot* genannt und muss als solche mit dem in Qt eingeführten Zugriffsmodifikator *slot* deklariert werden.

Zur Verbindung der Signale mit den Slots wird das Qt-Konstrukt *connect()* verwendet. Weitere Informationen finden sich in der Dokumentation zu Qt [26].

2.2.2 Kommunikationsmechanismus des USB-Moduls

Wie schon erwähnt, findet die Kommunikation über die Programmbibliothek *libusb* statt. Sie bietet für den synchronen Control-Transfer Modus die Funktion *libusb_control_transfer*.

Eine detaillierte Beschreibung der Funktionen von *libusb* finden sich in dessen Dokumentation [25]. Die Klasse *USBDevice* stellt eine vereinfachte Benutzung dieser Funktionen in C++ bereit. Grundsätzlich wird jede Kommunikation zwischen den USB-Modulen und dem Steuerungsrechner mithilfe von folgenden Methoden ermöglicht. Sie sind einfache Wrapper der Funktionalitäten von *libusb*:

- void USBDevice::**writeControlTransfer**(uint8_t request, uint16_t value, uint16_t index)
- void USBDevice::**writeControlTransfer**(uint8_t request, uint16_t value, uint16_t index, const std::vector<unsigned char>& data)
- std::vector<unsigned char> USBDevice::**readControlTransfer**(uint8_t request, uint16_t value, uint16_t index, uint16_t expected_data_length)
- std::vector<unsigned char> USBDevice::**readControlTransfer**(uint8_t request, uint16_t value, uint16_t index, uint16_t expected_data_length, int timeout)

Die Parameter haben folgende Bedeutung:

request: ID der anzusprechenden implementierten Firmware Funktion am USB-Modul z.B. `VENDOR_REQUEST_LASER_SET_SERVO`. Sie stellt das zweite Byte im Nutzdatenbereich des Setup-Pakets eines *USB-Control-Transfers* dar.

value: Ein Wert, der als Parameter an die mit **request** ausgewählte USB-Funktion übergeben wird. Er entspricht den dritten und vierten Byte (16 Bit) im *Transfer-Block*. Die Bedeutung dieses Werts muss für jede ID separat festgelegt werden.

index: Ein weiterer Parameter, der an die USB-Funktion übergeben wird. Bytes 5 und 6 (16 Bit) im *Transfer-Block*. Die Bedeutung wird wiederum individuell für jede Anforderung festgelegt.

data: Ein Datenarray, das an die anzusprechende USB-Funktion überreicht. Mittels Lesezugriff angeforderte Daten werden über den Rückgabewert der Funktionen übergeben.

expected_data_length: Die erwartete Anzahl von Bytes, die empfangen werden sollen.

timeout: Hier kann eine Zeit in Millisekunden übergeben werden, die gewartet wird, bevor die Methode den Empfang abbricht und zurückkehrt.

Die Methoden **readControlTransfer** geben das Ergebnis der Operation über ein Datenvektor `std::vector<unsigned char>` zurück.

2.2.3 Kamera Modul *FTCamera*

Aufgabe des Kamera-Moduls ist die Herstellung der Kommunikation sowie die Bedienung der Kamera. Die Kamera ORCAII-BT-512G (Hamamatsu) [16] ist DCAM (IIDC) (1394-based Digital Camera Specification) [17] kompatibel und wird mithilfe der Bibliotheken des Open-Source Projekts *libdc1394* [2] in das System integriert. Es handelt hierbei um eine Shared-Library (.so), die zur Laufzeit vorhanden sein muss und eingebunden wird. Die Klasse *FTCamera* (Abbildung 9) verwaltet diese Kommunikation und stellt sie als Ressource den beiden Klassen *FTImage* (2.2.13) und *ImageFetcher* zur Verfügung.

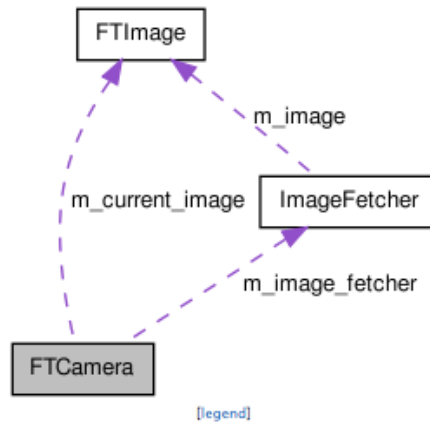


Abbildung 9: Kollaborationsdiagramm von FTCamera

2.2.3.1 Kommunikation

Durch die Kompatibilität mit DCAM (IIDC) können die entsprechenden Standardkommandos benutzt werden.

Eine Besonderheit der Kamera stellt die Auswahl des Video Modus (Tabelle 3) dar. Sie ermöglicht Aufnahmen in einem High-Speed-Modus und einem High-Precision Modus. Beide Modi ermöglichen jeweils ein Binning von 1x1, 2x2, 4x4 und 8x8 Pixel. Diese Einstellungen werden über das Standardkommando *dc1394_video_set_mode* wie folgt zugeordnet:

Tabelle 3: Zuordnung der Kamera Modi zu den Standard Video Modi nach IIDC (DCAM)

Kamera Modus		IIDC Video Modus Index
Readout	Binning	
High Speed (12Bit)	1x1	0 (DC1394_VIDEO_MODE_FORMAT7_0)
High Speed (12Bit)	2x2	1 (DC1394_VIDEO_MODE_FORMAT7_1)
High Speed (12Bit)	4x4	2 (DC1394_VIDEO_MODE_FORMAT7_2)
High Speed (12Bit)	8x8	3 (DC1394_VIDEO_MODE_FORMAT7_3)
High Precision (16Bit)	1x1	4 (DC1394_VIDEO_MODE_FORMAT7_4)

High Precision (16Bit)	2x2	5 (DC1394_VIDEO_MODE_FORMAT7_5)
High Precision (16Bit)	4x4	6 (DC1394_VIDEO_MODE_FORMAT7_6)
High Precision (16Bit)	8x8	7 (DC1394_VIDEO_MODE_FORMAT7_7)

2.2.4 Positionierungsmodul *FTPositioning*

Die Kommunikation des Koordinatentisches mit dem Steuerungsrechner findet über das RS232-Protokoll statt. Es werden vier COM-Ports benötigt. Standardmäßig stellt Linux die Verwendung von vier Ports zu Verfügung.

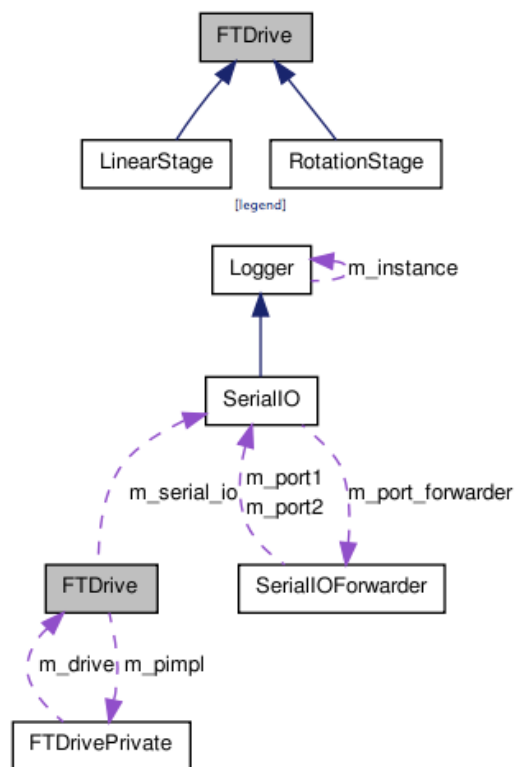


Abbildung 10: Kollaborationsdiagramm von *FTPositioning*

Am Motherboard des Steuerungsrechners befindet sich ein weiterer Port, wodurch die Anzahl der standardmäßig verfügbaren Ports überschritten wird. Um die Unterstützung von mehr als vier Ports unter Linux zu aktivieren, wird folgender Kernel-Boot-Parameter benutzt:

8250.nr_uaarts=7

Dieser Parameter teilt dem Kernel mit, dass er sieben Einträge für serielle Schnittstellen in /dev erzeugen und verwalten soll.

Die softwaretechnische Struktur von *FTPositioning* (Abbildung 10) besteht grundsätzlich aus der generischen Klasse *FTDrive*. Sie stellt die grundlegenden Funktionen wie die Parametrierung der Kommunikation, Beschleunigung, maximale Geschwindigkeit und die Umsetzung der Motorkommandos in das Telegrammformat für die Motor-Kommunikation zur Verfügung. Sie benutzt zur Kommunikation die Klasse *SerialIO*, welche die Low-Level Kommunikation (auf Telegrammebene) mit den Schrittmotoren verwaltet. Von ihr werden die Klassen *LinearStage* und *RotationStage* abgeleitet. Diese übernehmen die Spezialisierung auf den Achsentyp. Darunter fällt u.a. die Umrechnung von Motorschritten in Weglängen bzw. Winkel. Eine *Logger*-Klasse protokolliert den gesamten Telegrammverkehr zwischen Steuerungsrechner und den Motoren in eine definierbare Log-Datei. Weitere Informationen zur Logger-Funktion der Positionierung finden sich in der Betriebsanleitung (siehe Anhang). Außerdem besteht die Möglichkeit über die Klasse *SerialIOForwarder* die Kontrolle des Motors der Objektrotation temporär an einen weiteren COM-Port zu übergeben. Dies ermöglicht die Steuerung des Motors durch das DAVID-Scanner Projekt.

Die Klasse *FTPositioning* fasst die Achsen des Koordinatentisches samt Objektrotation und Laserarm zu einer Klasse zusammen.

2.2.4.1 Kommunikation

Es werden die jeweiligen Telegramme aus dem Handbuch [23] der Schrittmotoren benutzt. Eine detaillierte Auflistung der verwendeten Kommandos findet sich in der Projektarbeit zur Positionierung [22].

2.2.5 Laser Modul mit Filtrrad

Die Verwaltung des Lasers und der Filtrrades übernimmt die Klasse *FTLaser* (Abbildung 11). Die Verbindung zur Hardware wird von der Klasse *USBDevice* realisiert. Zusätzlich wurde ein Sicherheits-Heartbeat implementiert, der den Laser deaktiviert, wenn das Hauptprogramm beendet wird oder das System abstürzt.

Hierzu sendet die Klasse *LaserSafetyHeartbeat* periodisch Freigabesignale an die Laserhardware.

Diese Signale müssen innerhalb einer Torzeit von ca. 500 ms gesendet werden, um den Laser aktiviert zu halten.

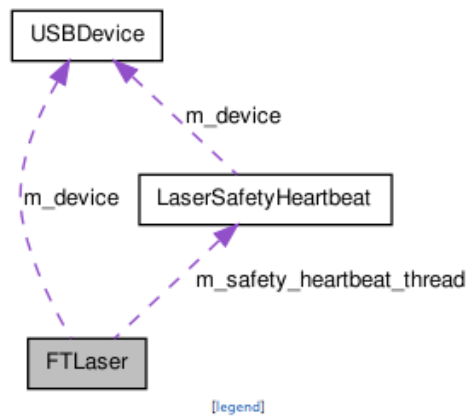


Abbildung 11: Kollaborationsdiagramm von FTLaser

Zusätzlich wird die Steuerung und die Justierung der Abschwächer Einheit bzw. des Filterrads von der Klasse *FTLaser* verwaltet.

2.2.5.1 Kommunikation

Zur Identifizierung des Moduls wird folgende ID benutzt:

LASER_PROJECT_ID = 0xF008

Die gewünschte Funktion wird über den *libusb* Parameter **request** übergeben (siehe Kapitel 2.2.2). Es werden folgende Funktionen für die Steuerung der Lasermoduls bzw. des Filterrades benutzt:

Tabelle 4: Übersicht der Funktionen und Parameter für die Kommunikation mit dem Laser Modules

request	Beschreibung
VENDOR_REQUEST_LASER_SET_SERVO=0x01	Setzt die Position des Filterrad-Servos auf value . Gültige Werte hierfür liegen im Intervall [0x9090,0x9A00].
VENDOR_REQUEST_LASER_GET_SERVO=0x02	Gibt die Position des Filterrad-Servos zurück. Sie steht im Datenvektor im 16 Bit Format (little endian)

VENDOR_REQUEST_LASER_SET_LASERATT= 0x03	Stellt die Abschwächung (die Position des Filterrades) auf die Stufe value . Gültige Werte hierfür liegen im Intervall [0,5]. Sie entsprechen den Abschwächungsstufen.
VENDOR_REQUEST_LASER_WRITE_SERVOVAL=0x04	Liest die aktuelle Position des Liferrad-Servos aus und schreibt sie als Abschwächung value in das EEPROM des Controllers.
VENDOR_REQUEST_LASER_SET_ONOFF=0x03	Aktiviert (value > 0) oder deaktiviert (value = 0) den Laser.
VENDOR_REQUEST_LASER_GET_ONOFF= x06	Gibt den aktuellen Betriebszustand des Lasers zurück. Das Ergebnis steht im Datenvektor, Byte[0], Bit[0].
VENDOR_REQUEST_LASER_GET_LASERATT=0x07	Gibt die aktuell eingestellte Abschwächung des Lasers zurück. Das Ergebnis (Stufe) steht im Datenvektor, Byte[0] und liegt im Intervall [0,5].
VENDOR_REQUEST_LASER_HEARTBEAT=0x08	Setzt den internen Heartbeat-Zähler der Laser-Hardwarekomponente zurück. Dieses Kommando muss mindestens alle 500ms gesendet werden, damit der Laser eingeschaltet bleibt. Dabei ist der Wert von value gleichgültig.

2.2.6 Galvo Modul

Die Klasse *FTGalvo* stellt Funktionen für die Positionierung des Laserstrahls mithilfe des Galvo-Spiegelsystems zur Verfügung. Sie benutzt ebenfalls die Klasse *USBDevice* für die Kommunikation mit der Hardware [19].

2.2.6.1 Kommunikation

Zur Identifizierung des Moduls wird folgende ID benutzt:

GALVO_PROJECT_ID = 0xF001

Die gewünschte Funktion wird über den *libusb* Parameter **request** übergeben (siehe Kapitel 2.2.2). Es werden folgende Funktionen für die Steuerung des Galvo-Scanners benutzt:

Tabelle 5: Übersicht der Funktionen und Parameter für die Kommunikation mit dem Galvo-Scanner

request	Beschreibung
VR_SEND_DAC = 0x03	Setzt die XY-Position Galvo-Scanners auf value=X und index=Y . Dabei entspricht das Intervall für Azimut [-32768,32767] und für Elevation [-32768,32767] dem Aussteuerungsbereich des Galvos.

Weitere (hier nicht benutzte) Funktionen finden sich in der verfassten Arbeit zum Modul [19].

Außerdem übernimmt *FTGalvo* auch die Umrechnung zwischen Winkel und DAC-Werten. Dabei werden Azimut- und Elevations-Winkel wie folgt umgerechnet:

$$x = k_x * \varphi + d_x \quad (2)$$

$$y = k_y * \theta + d_y \quad (3)$$

Die Parameter haben folgende Bedeutung:

x, y die DAC-Werte für Azimut und Elevation die zum Galvo-Modul übertragen werden

φ, θ die Winkel für Azimut und Elevation in Grad

k_x, k_y die Skalierungsfaktoren für Azimut und Elevation

d_x, d_y die Offsetwerte für Azimut und Elevation

2.2.7 Shutter Modul

Es stehen zwei Mechanismen zur Verfügung, um die Fotokathode des Bildverstärkers vor zu hoher Lichtintensität zu schützen:

- Es wird eine **Negativliste** mit verbotenen Positionen **vor** Beginn eines Messablaufs erzeugt. Sie wird durch Ausführen des gesamten Messablaufs mit geschlossenem Shutter generiert, indem Positionen, bei denen die Solarzelle des Shutters eine zu hohe Lichtintensität (excess-light) erfasst, in diese Liste eingetragen werden. Bei der anschließenden „normalen“ Ausführung der Messung bleibt der Shutter an jenen potentiell gefährlichen Positionen bei Laseraktivität geschlossen.
- Die Prüfung der Lichtintensität (excess-light) erfolgt vor **jeder** Laseraktivität bzw. Bilderfassung **während** des Messablaufs (on-the-fly). Dabei wird der Inhalt der Negativliste ignoriert.

Zur Erfassung des excess-light-Zustands wird das hardwareseitig aufbereitete Signal von der Solarzelle am Shutter-Arm benutzt. Dieses Signal wird periodisch von der Klasse *ShutterStateFetcher* abgefragt.

Sie greift auf die Klasse *FTShutter* (Abbildung 12) zu, welche die Bewegung des Shutter-Arms, die Behandlung des excess-light-Zustands sowie die Steuerung der Hintergrundbeleuchtung übernimmt. Wieder wird die Kommunikation über *USBDevice* abgehandelt.

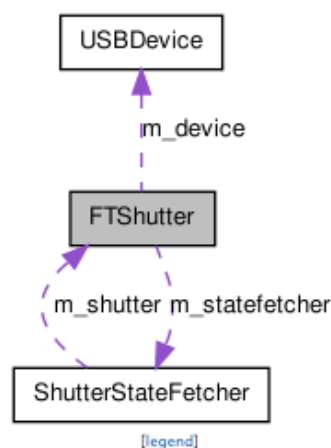


Abbildung 12: Kollaborationsdiagramm von *FTShutter*

2.2.7.1 Kommunikation

Zur Identifizierung des Moduls wird folgende ID benutzt:

SHUTTER_PROJECT_ID = 0xF009

Die Funktion wird über den *libusb* Parameter **request** übergeben. Es werden folgende Funktionen für die Steuerung Shutter Moduls benutzt:

Tabelle 6: Übersicht der Funktionen und Parameter für die Kommunikation mit dem Shutter

Request	Beschreibung
VENDOR_REQUEST_SHUTTER_SET_SHUTTER=0x01	Setzt den Betriebszustand des Shutter (geöffnet/geschlossen) auf value . (0 = geschlossen, 1 = geöffnet)
VENDOR_REQUEST_SHUTTER_GET_SHUTTER=0x02	Gibt den Betriebszustand des Shutters zurück. Dieses Kommando ist veraltet und wurde von Kommando VENDOR_REQUEST_SHUTTER_GET_SHUTTER_STATE abgelöst.
VENDOR_REQUEST_SHUTTER_GET_SHUTTER_STATE=0x05	Gibt den aktuellen Betriebszustand des Shutters sowie den Excess-Light-Status der Solarzelle zurück. Das Ergebnis steht im Datenvektor, Byte[0]. Dabei werden folgende Zustände bitkodiert OR-verknüpft: FTSHUTTER_EXCESS_LIGHT=0x80 FTSHUTTER_STATE_OPEN=0x01
VENDOR_REQUEST_SHUTTER_WRITE_SERVOVAL=0x06	Liest die aktuelle Position des Shutter-Servos aus und schreibt sie als Zustand value (0 = geschlossen, 1 = geöffnet) in das EEPROM des Controllers.
VENDOR_REQUEST_SHUTTER_SET_SERVO=0x07	Setzt die Position des Shutter-Servos auf value . Gültige Werte hierfür liegen im Intervall [0x9090,0x9A00].

2.2.8 Synthesizer (DDS)

Frequenz und Phase der beiden DDS-Kanäle verwaltet die Klasse *FTSynthesizer* (Abbildung 13). Sie benutzt zur Kommunikation mit dem DDS-Board die Klasse *USBDevice*. Eine Besonderheit des DDS-Bords stellt die Initialisierung der Phase dar. Die beiden Kanäle haben nach einem Initialisierungskommando einen willkürlichen Phasenversatz. Dieser kann, wenn benötigt, durch einen Offset korrigiert werden. Diese und andere Funktionen stellt die Klasse *FTSynthesizer* zur Verfügung. Weitere Informationen finden sich in der Projektarbeit zum DDS-Board [18].

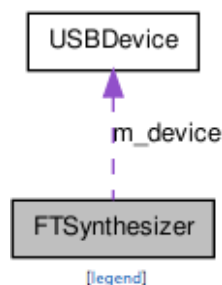


Abbildung 13: Kollaborationsdiagramm von *FTSynthesizer*

2.2.8.1 Kommunikation

Zur Identifizierung des Moduls wird folgende ID benutzt:

DDS_PROJECT_ID = 0xF005

Die gewünschte Funktion wird über den *libusb* Parameter **request** übergeben (siehe Kapitel 2.2.2). Es werden folgende Funktionen für die Steuerung DDS Moduls benutzt:

Tabelle 7: Übersicht der Funktionen und Parameter für die Kommunikation mit dem DDS Modul

Request	Beschreibung
VR_SET_LED=0x01	Aktiviert (value =1) oder deaktiviert (value =0) die Board-LED (an PORTC.1)
VR_TOGGLE_PIN=0x02	Aktiviert bzw. deaktiviert alternierend den durch value angegebenen Controller-Port mit dem Bitmuster in index . Dabei wird der Port wie folgt ausgewählt: Port A = 0x0A, Port B=0x0B, Port C=0x0C.

VR_SPI_TRANSMIT=0x03	Mit diesem Kommando wird eine Art Gateway zu den – über SPI – angeschlossenen DDS-Chips bzw. zur Takterzeugung realisiert. Die Initialisierung der Chips kann somit direkt auf Steuerungsrechner Ebene stattfinden.
VR_READ_PORT=0x04	Liest den Zustand aller drei Ports (A,B,C) ein und übergibt ihn im Datenvektor (3 Bytes, Byte[0]=Port A, Byte[1]=Port B, Byte[2]=Port C).
VR_READ_AD=0x05	Startet die AD-Wandlung eines Kanals und gibt das Ergebnis im Datenvektor zurück.

2.2.9 Backlight Modul

Die Klasse *FTBackLight* (Abbildung 14) stellt eine Ausnahme im Hardware-Software Korrespondenzkonzept des Systems dar. Da das Shutter-Modul noch genügend Ressourcen frei hatte, wurde die Ansteuerung der Hintergrundbeleuchtung vom Shutter-Modul übernommen. Daher liegt keine eigene Hardware-Komponente für die Beleuchtung vor. Sie wird über die Shutter-Komponente mitgesteuert. Daher greift sie direkt auf die Funktionen der Klasse *FTShutter* zu.

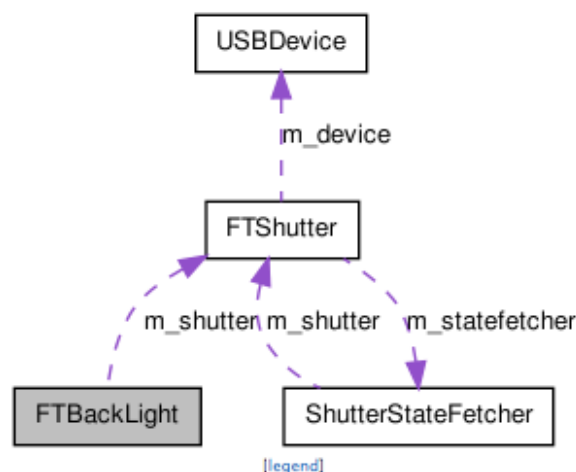


Abbildung 14: Kollaborationsdiagramm von *FTBackLight*

Weiters wurden Funktionen implementiert, um die Zündeigenschaften von Kaltkathodenröhren zu berücksichtigen. Durch die sogenannte „Smart Ignition“

Funktion wird sichergestellt, dass auch bei gering eingestellten Helligkeiten ein Zünden der Röhre möglich ist. Dies wird erreicht, indem zunächst die volle Versorgungsspannung an die CCFL gelegt wird und danach die Spannung auf den eingestellten Wert heruntergedimmt wird.

2.2.9.1 Kommunikation

Da die Backlight-Funktionalität hardwaremäßig das Shutter Modul übernimmt, wird zur Identifizierung die ID des Shutter Moduls benutzt:

SHUTTER_PROJECT_ID = 0xF009

Die gewünschte Funktion wird über den *libusb* Parameter **request** übergeben (siehe Kapitel 2.2.2). Es werden folgende Funktionen für die Steuerung Shutter Moduls benutzt:

Tabelle 8: Übersicht der Funktionen und Parameter für die Kommunikation mit der Hintergrundbeleuchtung

Request	Beschreibung
VENDOR_REQUEST_SHUTTER_SET_LIGHT_LEVEL=0x03	Setzt den PWM-Helligkeitswert der Hintergrundbeleuchtung auf value . Dabei muss dieser Wert im Intervall [0,255] liegen.

2.2.10 Synchrondemodulation

Eine der Hauptaufgaben des Steuerungsprogramms *ftcontrol* ist die Erfassung von Phasenbildern mithilfe der Kamera. Diese werden durch optische Synchrondemodulation des von der Probe kommenden Fluoreszenz- bzw. Streulichts erzeugt.

Der für die Anregung verwendete Laser wird um seinen Arbeitspunkt mit einem HF-Signal moduliert, das von einem Synthesizer-Kanal generiert wird. Der zweite Synthesizer-Kanal wird über eine passende HF-Endstufe mit dem Bildverstärker verbunden. Die Ansteuerung erfolgt dabei so, dass die Verstärkung des Bildverstärkers ebenfalls mit der HF moduliert wird. Der so realisierte synchrone Verschlussmechanismus kann zur Demodulation verwendet werden indem der Phasenversatz beider Synthesizer-Kanäle verändert wird.

Es werden also grundsätzlich zwei Aufnahmen benötigt. Zuerst wird eine Aufnahme ohne Phasenversatz zwischen den beiden HF-Kanälen erfasst (In-Phase-Komponente I), anschließend mit einem Versatz von 90° zueinander (Quadratur-Komponente Q_+). Praktisch kann auf einen Nullphasenabgleich der beiden HF-Kanäle verzichtet werden, wenn eine dritte Aufnahme der Probe mit -90° Versatz im Bezug auf die zuvor erfasste ‚Null‘-Phasenlage durchgeführt wird (Komponente Q_-). Dieser Ansatz bzw. Ablauf wird vom Steuerungsprogramm *ftcontrol* verwendet. Die hierzu verwendeten Kalkulationen lauten wie folgt:

Bezeichnet man die In-Phasen und Quadraturbilder mit I , Q_+ und Q_- , so kann man Real- und Imaginärteil des Bildes *image* aus

$$\text{Re}(\text{image}) = 2 * I - Q_+ - Q_- \quad (4)$$

$$\text{Im}(\text{image}) = Q_- - Q_+ \quad (5)$$

berechnen.

Eine Herleitung findet sich in Kapitel 0. Bei der internen Verwendung und Weitergabe von Bildern wird die Klasse *FTImage* (2.2.13) benutzt. Sie bietet einen Shared-Data Bereich der bei der Übergabe der Klasse *by-reference* mitübergeben wird. Dies verhindert eventuelle Überlaufprobleme des Heap-Speichers.

2.2.11 Image Viewer

Zur Darstellung und Analyse der erfassten Bilddaten wird im Hauptprogramm *ftcontrol* der Image Viewer benutzt. Er bietet die Möglichkeit, Bilder in verschiedenen Falschfarbendarstellungen zu betrachten sowie durch Selektieren von Bereichen im Bild (sogenannte ROI's), numerische bzw. statistische Kennzahlen zu berechnen. Er wird in der Klasse *FTImageView* implementiert und setzt sich aus den Komponenten *FTImageGraphicsView* und *FTImageScene* zusammen. Die Klasse *FTImageView* wird im Kamera-Dialog und als eigenständiger Viewer benutzt (vgl. Abbildung 15).

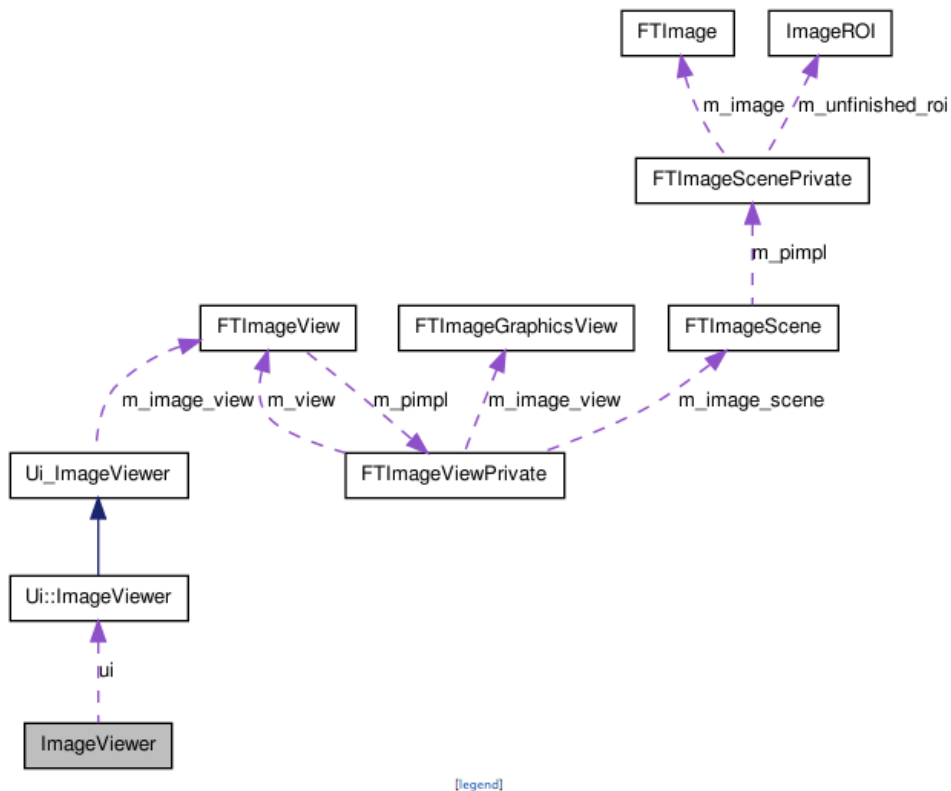


Abbildung 15: Kollaborationsdiagramm von ImageViewer

2.2.12 Sicherheits-Watchdog

Da sich bei der Verwendung des für das Fluoreszenztomographie-System benutzten Lasers ein potentiell Sicherheitsrisiko bzw. Risiko der Beschädigung von Geräten ergibt, werden spezielle Maßnahmen ergriffen, um den Umgang mit dem Laser möglichst sicher zu gestalten. Diese Maßnahmen werden durch Mechanismen implementiert, von denen die softwaretechnischen Lösungen in der Klasse *Watchdog* abgebildet werden. Hierzu zählen:

- **Sicherheitssperrung der Laseraktivierung bei Positionsänderung:** Die Aktivierung des Lasers wird nur dann erlaubt, wenn keine Positionsänderung mittels Koordinatentisch, Laserarm oder Objektrotation stattfindet. Dies verhindert eine eventuelle Überbelichtung des MCP-Bildverstärkers durch reflektierte oder direkte Laserstrahlung.
- **Aktivierung des Shutters bei Positionsänderung:** Wird die Option „On-The-Fly-Excess light check“ (siehe Betriebsanleitung im Anhang) benutzt, so steuert die Klasse *Watchdog* die Aktivierung des Shutters **vor** jeder

Positionsänderung mittels Koordinatentisch, Laserarm oder Objektrotation. Daraufhin wird durch Auswerten des *Excess-Light* Signals eine potentielle Überbelichtung des MCP-Bildverstärkers durch den Laser ermittelt. Ist keine Gefährdung erkennbar, wird der Shutter wieder deaktiviert (geöffnet).

Die Klasse *Watchdog* greift somit auf die Module *FTPositioning*, *FTLaser* und *FTShutter* zu (Abbildung 16).

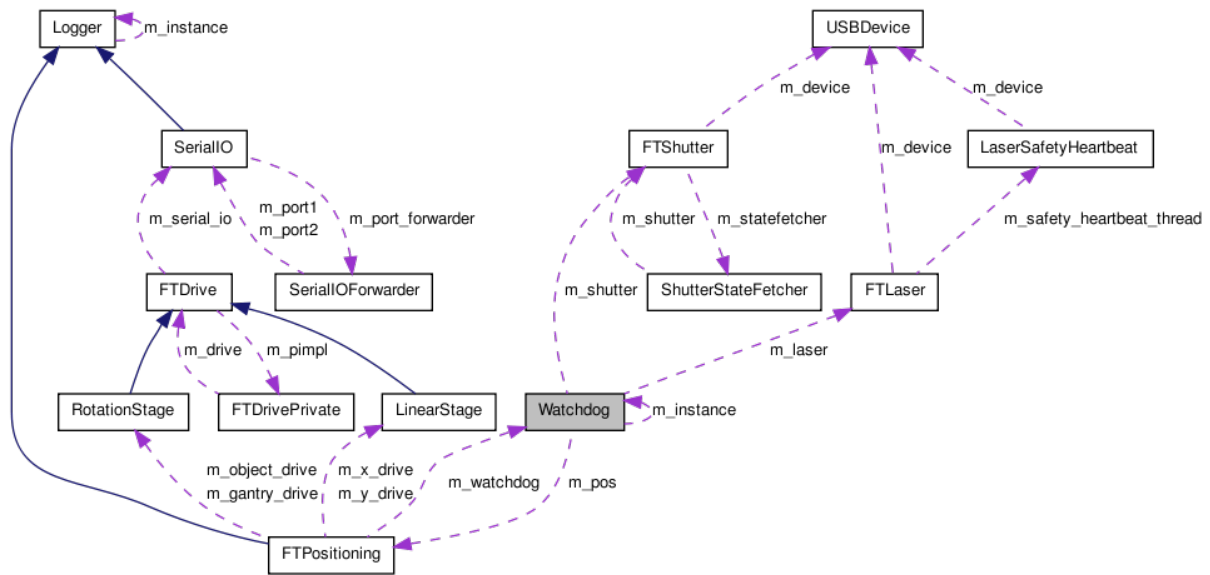


Abbildung 16: Kollaborationsdiagramm von Watchdog. Sie hat auf Shutter, Positionierung und Laser Zugriff

2.2.13 Bilddatencontainer FTImage

Die Verwaltung der Bilddaten sowohl innerhalb des Hauptprogramms *ftcontrol* als auch zur Sicherung bzw. Weiterverwendung wird durch die Klasse *FTImage* realisiert. Sie stellt einen Bilddaten-Container sowie einen Metadaten- bzw. Protokoll-Container für diverse Parameter wie z.B. Belichtungszeit bei Erfassung zur Verfügung. Weiters werden grundlegende Bildverarbeitungs- und Analysefunktionen wie z.B. *minValue*, *maxValue* sowie Operatorüberladungen von ***, *+* und *,* zu komfortablen Auswertung und Manipulation der Bilddaten angeboten.

Zur Sicherung der Bilddaten wird ein eigenes Dateiformat (.FTI), basierend auf XML verwendet. Dies ermöglicht eine einfache Handhabung der Bilddaten zur Weiterverarbeitung in anderen Programmen wie z.B. Matlab.

Für Matlab steht ein Import-Tool zur Verfügung welches FTI-Dateien importiert und als Vektor zur Verfügung stellt. Es wurde im Rahmen einer Projektarbeit [12] entwickelt. und wird im Ordner *control/matlab* zur Verfügung gestellt.

2.2.14 Script Engine

Für die Realisierung und Ausführung der Messprotokolle wird die Qt-eigene Scripting-Funktionalität *QScript* benutzt. Sie basiert auf JavaScript [32] und stellt somit eine mächtige und flexible Lösung für diese Aufgabe dar.

Basis des Scripting-Mechanismus ist die Klasse *QScriptEngine*. Sie stellt sämtliche Funktionen zur Verfügung welche zur Evaluierung von JavaScript-Programmen notwendig sind.

Um einen komfortablen Zugang zu sämtlichen Ressourcen des Tomographiesystems zu gewährleisten, wird die Klasse *ScriptMachine* eingeführt, welche alle – für die vollständige Ausführung von Messprotokollen – benötigten Softwaremodule als Referenzen verwaltet und zur Skriptausführung zur Verfügung stellt. Eine komplette Darstellung der Ressourcen zeigt das Kollaborationsdiagramm der Klasse *ScriptMachine* (Abbildung 17).



Abbildung 17: Kollaborationsdiagramm von ScriptMachine. Durch die Verwendung sämtlicher Ressourcen in der Klasse wird in diesem Diagramm nahezu das gesamte Steuerungsprogramm ftcontrol abgebildet.

2.2.15 Konsole

Zur Messprotokoll-Entwicklung bzw. zum Testen von Befehlen gibt es die Möglichkeit, über eine – in das Hauptfenster integrierte – Konsole Skript-Befehle direkt auszuführen. Diese Funktionalität stellt die Klasse *Console* (Abbildung 18) zur Verfügung und wurde von Manuel Freiberger entwickelt. Sie bietet zahlreiche Funktionen zur komfortablen Entwicklung von Skripten.

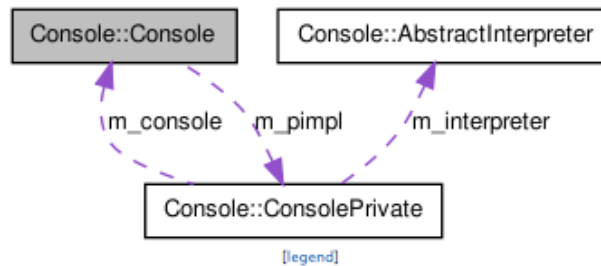


Abbildung 18: Kollaborationsdiagramm von Console

2.2.16 Logger

Bei der Ausführung eines Messablaufs werden über die Klasse *Logger* (Abbildung 19) Informationen über die Positionierung im Messablauf aufgezeichnet. Diese Informationen werden direkt von vom Modul *FTPositioning* erzeugt. Sie werden in der Logger-Datei gespeichert. Diese Datei kann im Positionier-Dialog unter *Settings* eingestellt werden (siehe Betriebsanleitung im Anhang).

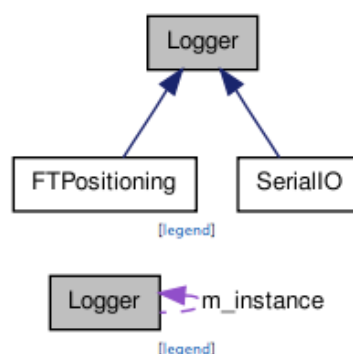


Abbildung 19: Kollaborationsdiagramm von FTLogger

2.2.17 Systemkonfiguration

Das Hauptprogramm *ftcontrol* benutzt zur Speicherung von Systemparametern und Einstellungen die Klasse *QSettings* von *Qt*. Es handelt sich dabei um eine Sicherungsmöglichkeit dieser Parameter im INI-Format (vgl. typische Windows Initialisierungsdateien).

Es besteht die Möglichkeit, diese Systemkonfiguration zu sichern bzw. eine neue oder andere Konfiguration in das Hauptprogramm einzuspielen. Weiters besteht die Möglichkeit, die Standardkonfiguration wiederherzustellen. Weitere Informationen finden sich in der Betriebsanleitung.

2.3 Risikoanalyse

Im Rahmen dieser Diplomarbeit wurde eine Risikoanalyse des gesamten Fluoreszenztomographie-Systems durchgeführt. Zweck einer Risikoanalyse ist die Ermittlung bzw. Bewertung von möglichen Gefahren, die vom Fluoreszenztomographie-System ausgehen können, um entsprechende Maßnahmen zur Vermeidung bzw. Reduktion dieses Gefahrenpotentials treffen zu können. Da das Fluoreszenztomographie-System nicht für die Anwendung am Menschen entwickelt wurde kann die Maschinenrichtlinie [33] zur Bewertung der Risikofaktoren des Systems herangezogen werden.

Unter Zuhilfenahme der Europäischen Norm EN 60601-1 für medizinische elektrische Geräte [34] sowie aufbauend auf einer existierenden Analyse in der Diplomarbeit von Michael Kalis [35] wurden für dieses System bestehende sowie hinzugekommene Sicherheitsaspekte und Risikotabellen erneut betrachtet und eingeschätzt.

Sie beinhalten eine Auflistung aller möglichen Gefahren, die bei der Anwendung im Normalfall (NC) und im ersten Fehlerfall (SFC) auftreten können.

Für die Eintrittswahrscheinlichkeiten (E) gilt eine 6-stufige Einteilung:

- | | |
|--------------------|----------------|
| 1 unglaublich | 4 gelegentlich |
| 2 unwahrscheinlich | 5 manchmal |
| 3 selten | 6 häufig |

Die Einteilung der möglichen Folgen (F) bei genannten Eintrittswahrscheinlichkeiten:

- | | |
|----------------|---|
| 1 gering | Keine bzw. geringe Verletzungen |
| 2 mittel | Verletzungen |
| 3 schwer | schwere Verletzungen, Tod |
| 4 katastrophal | schwere Verletzungen bei mehr als einer Person, Tod |

Die Kombination der möglichen Folgen bei einer bestimmten Eintrittswahrscheinlichkeit ergibt folgende Aufstellung:

Tabelle 9: Einteilung der Risikostufen für die entsprechenden Folgen und deren Eintrittswahrscheinlichkeiten

Eintrittswahrscheinlichkeit (E)	Folgen (F)			
	gering (1)	mittel (2)	schwer (3)	katastrophal (4)
unglaublich (1)	IV	IV	IV	IV
unwahrscheinlich (2)	IV	III	III	III
selten (3)	III	III	III	II
gelegentlich (4)	III	III	II	I
manchmal (5)	III	II	I	I
häufig (6)	II	I	I	I

Die verwendeten Risikogruppen (RS) haben dabei folgende Bedeutung:

- I inakzeptables Risiko
 (muss durch Reduzieren der Eintrittswahrscheinlichkeit oder der Schwere des Schadens verringert werden)
- II unerwünschtes Risiko
 (tolerierbar, wenn die Vermeidung bzw. die Kosten in keinem Verhältnis zum Nutzen stehen)
- III tolerierbares Risiko
 (kann aus wirtschaftlichen Gründen in Kauf genommen werden)
- IV vernachlässigbares Risiko
 (keine weiteren Maßnahmen erforderlich)

Risiken, welche nicht in die Gruppen III oder IV fallen müssen durch geeignete Maßnahmen auf ein akzeptables Maß reduziert werden. Hierzu werden folgende Schutzmaßnahmen nach aufsteigenden Rang in Sicherheitsstufen (SS) eingeteilt:

- 1 konstruktive und technische Maßnahmen
- 2 organisatorische Maßnahmen, Zugangsbeschränkungen
- 3 schriftliche Hinweise, Warnschilder, Schulungen

Anschließend werden die durch die Schutzmaßnahmen reduzierten Risiken noch einmal in Bezug auf Eintrittswahrscheinlichkeit und Folgen bewertet und einer Restrisikostufe (RSS) zugeordnet.

Werden die Schutzmaßnahmen wieder als unzureichend eingestuft, sind weitere Maßnahmen erforderlich, um das Restrisiko auf ein akzeptables Maß zu senken.

2.4 Ableit- und Berührungsstrommessung

Beim Fluoreszenztomographiesystem handelt es sich um ein Gerät mit festem Versorgungsanschluss, dessen Korpus aus isolierenden mitteldichten Faserplatten (MDF) besteht. Unter Zuhilfenahme der Europäischen Norm EN 60990 (Verfahren zur Messung von Berührungsstrom und Schutzleiterstrom) [36] wird eine Berührungsstrommessung und eine Ableitstrommessung durchgeführt. Zur Bewertung der menschlichen Spürbarkeit und Reaktion wird eine Impedanznachbildung (Abbildung 20) benutzt. Für realistische Bedingungen werden **alle** Komponenten des Systems (z.B. Steuerungsrechner, Kamera-Steuergerät) von der Hauptverteilerleiste mit Spannung versorgt. Um eine Verfälschung der Messung durch Leckströme über das LAN-Kabel zu vermeiden, wird dieses von der Netzwerkdose getrennt. Weiters wird der Schutzleiter an der Hauptverteilerleiste durchtrennt. Der Berührungsstrom wird durch Messung der Spannung U_1 erfasst. Mit

$$I_{Ber} = \frac{U_1}{500} \quad (6)$$

kann der bewertete Berührungsstrom I_{Ber} berechnet werden. Die Spannung wird mit einem Multimeter (Normameter MP13, Norma) ermittelt. Es werden folgende Konfigurationen gemessen.

- I Direkte Messung durch Verbinden des Tomographen-Schutzleiters mit der Schutzterde über die Impedanznachbildung (Klemmen A und B). Die Messung erfolgt einmal am eingeschalteten System und einmal mit durchtrenntem Nullleiter.
- II Messung an den, im geschlossenen Zustand des Tomographen berührbaren metallischen Teilen (Schlüsselschalter, Schrauben) durch Verbinden der Teile mit der Schutzterde über die Impedanznachbildung. Die Messung erfolgt einmal am eingeschalteten System und einmal mit durchtrenntem Nullleiter.
- III Messung am Holzgehäuse des Tomographen durch Anbringen einer Messingplatte (30x60mm ArtNr: 771-033, RS Components) an der Tomographentür. Diese wird über die Impedanznachbildung mit der Schutzterde verbunden. Die Messung erfolgt einmal am eingeschalteten System und einmal mit durchtrenntem Nullleiter.

Zweck einer Messung bei durchtrenntem Nullleiter ist die Ermittlung des maximalen Ableit- bzw. Berührungsstroms im Fehlerfall.

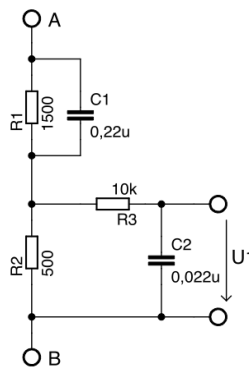


Abbildung 20: Impedanznachbildung mit Filter nach EN 60990

Zusätzlich wird der Ableitstrom des Fluoreszenztomographiesystems direkt, also ohne Impedanznachbildung, gemessen. Dabei wird der Schutzleiter an der Hauptverteilerleiste aufgetrennt und ein Amperemeter (Typ EX520, Extech) dazwischen geschaltet. Es wird der Ableitstrom folgender Last-Konfigurationen gemessen:

- V Messung **ohne** galvanische Verbindung zum Steuerungsrechner (USB Kabel) **und** zum Kamera-Steuergerät.
- VI Messung **ohne** galvanische Verbindung zum Steuerungsrechner jedoch **mit** angeschlossenem Kamera-Steuergerät.
- VII Messung des **gesamten** betriebsfähigen Systems.

Anschließend wird die gesamte Ableitkapazität ermittelt, welche durch die diversen Netzfilter im System zustande kommt. Die verwendete Ersatzschaltung für die Ableitkapazitäten [36] zeigt Abbildung 21.

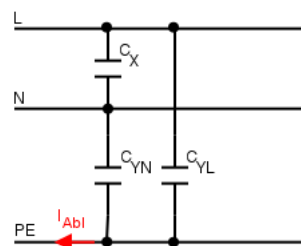


Abbildung 21: Kapazitive Komponenten eines Netzfilter

Die Kapazitäten C_{YN} und C_{YL} stellen die Entstör-Y-Kapazitäten der Netzfilter dar. Sie werden zu $C_{Y,Gesamt}$ zusammengefasst und bestimmen den Ableitstrom I_{Abl} , der sich nach

$$I_{Abl} = \omega \cdot C_{Y,Gesamt} \cdot U_{Netz} \quad (7)$$

$$\omega = 2 \cdot \pi \cdot f \quad (8)$$

mit $U_{Netz} = 230V$ und $f=50Hz$ ergibt. Durch Umstellen von Formel 7 ergibt sich eine Ableitkapazität des gesamten betriebsfähigen Systems (Konfiguration VII) von

$$C_{Y,Gesamt} = \frac{I_{Abl}}{\omega U_{Netz}} = \frac{1,21 \cdot 10^{-3}}{2 \cdot \pi \cdot 50 \cdot 230} = 16,7 \cdot 10^{-9} = 16,7nF \quad (9)$$

3 Ergebnisse und Diskussion

3.1 Risikoanalyse

Durch die Aufstellung der Risikotabellen auf den folgenden Seiten konnte bei den betrachteten Komponenten des Fluoreszenztomographie-Systems grundsätzlich ein akzeptables Restrisiko ermittelt werden.

Jedoch wurde auch erkannt, dass jedenfalls eine Unterweisung bzw. Schulung sowie eine Zugangsbeschränkung für die bedienenden Personen erforderlich ist, um das Restrisiko des gesamten Systems weiter zu senken.

Begriffsdefinitionen:

NC ... Normalbedingungen (Normal Condition) E ... Eintrittswahrscheinlichkeit (1-6) RS ... Risikostufe (I – IV)
 SFC ... Erster Fehlerfall (Single Failure Condition) F ... Folgen (1-4) RSS ... Restrisikostufe (I – IV)
 SS ... Sicherheitsstufe (1-3)

Gefahrenquelle	Gefahrenursache	NC	SFC	Mögliche Auswirkungen	E	F	RS	Abhilfemaßnahmen	SS	E	F	RRS
----------------	-----------------	----	-----	-----------------------	---	---	----	------------------	----	---	---	-----

Gefährdungen die mit dem Betrieb des Lasers verbunden sind												
Lasermodule	Laserstrahlung, hohe Lichtintensität	X		Bleibende Augenschäden	6	3	I	Abdeckungen, softwaretechn. Sicherung, Schutzbrillen, Gebotsschild, Einschulung, Zugangsbeschränkung, Schlüsselschalter, Verriegelung der Wartungsklappe, Türkontakt	1 2 3	3	1	III
Galvo-Scanner	Laserstrahlung, unkontrollierte Positionierung, hohe Lichtintensität	X	X	Bleibende Augenschäden	6	3	I	Abdeckungen, Schutzbrillen, Schlüsselschalter, Verriegelung der Wartungsklappe, Türkontakt	1 3	3	1	III
Filterrad	Deaktivierung der Abschwächung, hohe Lichtintensität	X	X	Bleibende Augenschäden	4	3	II	Softwaretech. Sicherungen, Schutzbrillen, Schulung, Schlüsselschalter, Verriegelung der Wartungsklappe, Türkontakt	1 3	3	3	III

Gefahrenquelle	Gefahrenursache	NC	SFC	Mögliche Auswirkungen	E	F	RS	Abhilfemaßnahmen	SS	E	F	RRS
----------------	-----------------	----	-----	-----------------------	---	---	----	------------------	----	---	---	-----

Funktionsfehler												
Reinigung	Falsche Reinigung der optischen Oberflächen	X		Zerkratztes Objektiv, schlechte Bildqualität	3	1	III	Reinigung laut Handbuch	3	1	1	IV
Kanten, Kabel und Gegenstände auf dem Boden	Abschaltung oder Ausfall der Raumbeleuchtung des abgedunkelten Raums	X	X	Stolpergefahr, Verletzungen bei Zusammenstößen	3	1	III	Manuelle Beleuchtung mittels Notleuchte	1	2	1	IV
Gefährdungen durch elektrische Energie												
Hochspannungsnetzteil	Isolationsfehler bei Leitungen und Kabel		X	Stromschlag	2	3	III	Geerdetes Metallgehäuse, weitere Isolierung	1	2	1	IV
	Freiliegende Kontakte der Hochspannungsanschlüsse bei Öffnen des Gehäuses		X	Stromschlag	5	3	I	Unterweisung, Arbeiten nur nach Abschalten, Erden und Kurzschließen des Geräts	1 2 3	2	1	IV
ATX-Netzteile	Funktionsfehler		X	Stromschlag	2	3	III	Arbeiten nur nach Abschalten des Geräts	1	2	1	IV

Gefahrenquelle	Gefahrenursache	NC	SFC	Mögliche Auswirkungen	E	F	RS	Abhilfemaßnahmen	SS	E	F	RRS
Mechanische Gefährdungen												
Messkammer	Umfallen des Gehäuses		X	Verschiedene Verletzungen	2	3	III	Stabile Lagerung mit Standfuß und geeigneten Tischbeinen	1	1	3	IV
Wartungsklappe	Herunterklappen der schweren Holzplatte	X	X	Kopfverletzungen, Quetschungen	6	3	I	Beidseitige Fixiervorrichtungen, Anheben der Klappe zu Zweit	1 3	3	3	III
Positioniereinheit	Bewegte Teile (Verschiebung und Rotation)	X		Quetschungen, Verletzungen	3	2	III	Warnschild	3	2	2	III

Gefahrenquelle	Gefahrenursache	NC	SFC	Mögliche Auswirkungen	E	F	RS	Abhilfemaßnahmen	SS	E	F	RRS
----------------	-----------------	----	-----	-----------------------	---	---	----	------------------	----	---	---	-----

Gefährdungen durch Chemikalien												
Methanol	Verschlucken, Hautkontakt, Bildung explosionsfähiger Gemische, Brandgefahr			Vergiftungen, Explosionsgefahr, Förderung von Bränden	4	3	II	Unterweisung, Handschuhe, Aufbewahrung im Chemieschrank, Belüftung	1 2 3	1	3	IV
Ethanol	Bildung explosionsfähiger Gemische, Brandgefahr			Explosionsgefahr, Förderung von Bränden	3	2	III	Unterweisung, Aufbewahrung im Chemieschrank	2 3	1	2	IV
Rhodamin 6G	Verschlucken, Hautkontakt			Kanzerogene Wirkung, schwach toxisch	4	2	III	Unterweisung, Handschuhe, Mundschutz, Schutzbrille, Aufbewahrung im Chemikalienschrank	2 3	1	2	IV
Gefährdungen durch Verbrennungsprodukte im Brandfall												
Bindemittel-Verbrennungsprodukte	MDF-Plattenkonstruktion			Gesundheitliche Schäden	2	4	III	Brandmelder, Belüftung	1	1	4	IV
HCl-Dioxine	Kabelisolierungen aus PVC			Gesundheitliche Schäden	2	3	III	Brandmelder, Belüftung	1	1	3	IV
HF-Verbindungen	Teflon			Gesundheitliche Schäden	2	3	III	Brandmelder, Belüftung	1	1	3	IV
Silan-Dämpfe	Silikonschläuche			Gesundheitliche Schäden	2	3	III	Brandmelder, Belüftung	1	1	3	IV

3.2 Ableit- und Berührungsstrommessung

Die Ergebnisse der Berührungsstrommessung zeigt Tabelle 10. Die Berührungsströme bei den Konfigurationen II bis IV konnten mit dem verwendeten Multimeter (Normameter MP13, Norma) nicht mehr gemessen werden, da sie unterhalb der Rauschspannung des Geräts lagen. In Tabelle 11 ist ersichtlich, dass bei Konfiguration VII (Verbindung mit Steuerungsrechner) ein maximaler (direkter) Ableitstrom von 1,21 mA auftritt.

Tabelle 10: Messungsergebnisse des bewerteten Berührungs- bzw. Ableitstroms

Konfiguration	U1 (gemessen) [mV]		bewerteter Berührungsstrom I_{Ber} (berechnet) [μ A]	
	eingeschaltet	durchtrennter Nulleiter	eingeschaltet	durchtrennter Nulleiter
I	2,7	3,6	5,4	7,2
II – IV	nicht messbar	nicht messbar	nicht messbar	nicht messbar

Tabelle 11: Messergebnisse der Ableitströme des Systems

Konfiguration	Ableitstrom [mA]
V	0,52
VI	0,56
VII	1,21

Gemäß der Europäischen Norm EN 50178 (Ausrüstung von Starkstromanlagen mit elektronischen Betriebsmitteln) liegt der maximale zulässige Wert des Ableitstroms für Geräte mit Steckeranschluss bei 3,5 mA. Wird dieser Wert überschritten, muss das Gerät über einen festen Anschluss verfügen, was beim Fluoreszenztomographiesystem der Fall ist. Dies verdeutlicht, dass bei ordnungsgemäßer Bedienung keine Gefahr von den auftretenden Ableit- bzw. Berührungsströmen (Tabelle 10, Tabelle 11) im Fehlerfall ausgeht. Die errechnete Ableitkapazität (Formel 9) spiegelt den Schachverhalt wider, dass im System vorwiegend Schaltnetzteile eingesetzt werden, deren Entstörmaßnahmen im Vergleich zu herkömmlichen Netzteilen ausgeprägter sind.

3.3 Betriebssoftware

Das Programm *ftcontrol* wurde erfolgreich realisiert und bietet alle geforderten Funktionen. Es liegt zur Zeit in der Version 1.5.05 (Stand 10.2.2012) vor und hat mittlerweile eine Größe von etwa 27.100 effektiven Programmzeilen in über 190 Dateien. Durch die Verwendung von Messprotokollen welche in *JavaScript* entworfen werden können, wurde ein hoher Grad an Flexibilität erreicht.

Das Programm ist voll funktionsfähig und wurde bereits für Messungen im Rahmen der Kooperation mit der Universität Aachen (Deutschland) eingesetzt. Es sollte jedoch entwicklungsstechnisch in das Beta-Stadium eingestuft werden, da aufgrund der Komplexität des Fluoreszenztomographie Systems an einigen Stellen evtl. noch Verbesserungspotential besteht. Aus diesem Grund existiert in der DoxyGen Dokumentation eine Todo-Liste mit Verbesserungsvorschlägen, welche zum Zeitpunkt der Verfassung dieser Arbeit jedoch grundsätzlich als „nice-to-have“ bewertet wurden. Ein Problem, welches sich durch einen – zum Zeitpunkt der Entwicklung der Software – noch nicht behobenen Fehler in der Script-Engine von Qt ergab ist das Beenden bzw. Abbrechen eines Messablaufs. Durch einen Fehler in der Zeitrechnung der Script-Engine wird eine Stop-Anforderung quasi ignoriert. Dies führt dazu, dass der Messablauf nicht abgebrochen werden kann. Die Beendigung erfolgt entweder selbstständig nach Ende des Messablaufs (was je nach Protokoll sehr lange dauern kann) oder durch Beenden des Hauptprogramms *ftcontrol*.

Bei der Entwicklung der Kameraschnittstelle wurde das erfasste Bildmaterial mit den Bilddaten der Kamera bei Ansteuerung mittels Steuerungssoftware des Herstellers Hamamatsu verglichen um etwaige Differenzen in der Parametrierung der Kamera festzustellen. Hierzu wurde jeweils ein Bild der gleichen Szene bzw. Beleuchtungsverhältnisse erfasst und voneinander subtrahiert. Resultat war erwartungsgemäß das Bildrauschen und einige wenige Artefakte welche mit hoher Wahrscheinlichkeit durch die leicht unterschiedliche Ausleuchtung bei den Aufnahmen entstanden.

3.4 Hardware

3.4.1 Allgemein

Der Bau der Hardware-Module wurde erfolgreich durchgeführt. Alle Module erfüllten die Anforderungen. Zusätzlich wurde beim **Lasermodule** die Möglichkeit geschaffen manuell d.h. am Modul durch zwei Tasten das Filterrad einzustellen ohne mit dem Steuerungsrechner verbunden zu sein. Durch die Aktivierung bzw. Deaktivierung des Lasers sowie der HF Ankoppelung durch HF-Relais auf der Platine hatte es anfangs einige Probleme mit der USB-Verbindung gegeben, welche sich jedoch durch Schirmen der Mikrocontrollerplatine vermindern ließen. Zusätzlich wurde zur Reduzierung der Störeinflüsse auf das Lasermodule eine Metall-Einhausung angefertigt.

Das **Shutter-Module** wurde ursprünglich von Gerald Schwarz im Rahmen einer Projektarbeit [12] entwickelt. Jedoch stellten sich die verwendeten CD-Laufwerke als sehr fehleranfällig heraus. Aus diesem Grund wurde eine neue Version des Shutters entwickelt, welche in dieser Arbeit dokumentiert ist. Durch den relativ großen Arm musste der verwendete Servo in bestimmten Situationen permanent ausregeln. Dies führte zu einer „Verunreinigung“ der Spannungsversorgung durch Spikes. Eine Entkopplung der Spannungsversorgung vom Rest des Systems sowie eine Deaktivierung des Servos nach einer Totzeit von einigen Sekunden konnte dieses Problem lösen.

Es stellte sich heraus, dass die **Kamera** von Hamamatsu [16] einen Linearitätsfehler im High-Precision Modus aufwies weshalb sie zur Reparatur bzw. Tausch geschickt wurde. Das zur Verfügung gestellte temporäre Ersatzgerät hatte leider das Problem schon bei geringen Helligkeiten „Blooming“ zu erzeugen. Erst gegen Ende dieser Arbeit konnte wieder auf eine funktionierende Kamera zugegriffen werden.

Alle übrigen entwickelten Module funktionierten von Anfang an klaglos. Jedoch wurde erkannt, dass eine Kommunikation via USB keine optimale Lösung bei den vorhandenen Leitungslängen sowie bei den gegebenen Störungspotential (HF, Hochspannung, Spikes durch Servos bzw. Galvos,...) darstellt.

3.4.2 USB Kommunikation

Ein großer Kritikpunkt am Fluoreszenztomographie System wurde in der USB Kommunikation zwischen den Hardware-Modulen und dem Steuerungsrechner erkannt. Durch die Leitungslängen bzw. Störeinflüsse stieß der USB-Standard an seine Grenzen. Dies äußerte sich vor allem in teils massiven Kommunikationsproblemen, sporadisches Rücksetzen der Hardware-Module oder durch unkontrollierte Aktivierung des Bootloader-Modus der Module. Lösung dieses Problems wäre beispielsweise eine Umstellung der Kommunikation zwischen den Teilnehmern auf einen robusteren Übertragungsstandard wie z.B. CAN oder Profibus. Dabei ist CAN im Allgemeinen vorzuziehen da es für diesen Standard sehr vielfältige Controller-Komplettlösungen am Markt gibt und dieser Feldbus mit einer maximalen Übertragungsrate von 1MBit leicht ausreichen würde.

4 Anhang

Literaturverzeichnis

- [1] Hamamatsu Photonics. [Online]. http://jp.hamamatsu.com/products/sensor-etd/pd014/V7090U-71/index_en.html
- [2] sourceforge.net. [Online]. <http://sourceforge.net/projects/libdc1394/>
- [3] M. Freiberger. IMT TU Graz. [Online]. <http://www.imt.tugraz.at/index.php/education/41/125-usb-bootloader>
- [4] microchip.com. [Online]. <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010280>
- [5] www.thorlabs.com. [Online]. <http://www.thorlabs.com/thorProduct.cfm?partNumber=NE510B>
- [6] Conrad Electronic. [Online]. <http://www.conrad.at/ce/de/product/230500/TOP-LINE-MINI-SERVO-ES-05-JR/?ref=search>
- [8] Timucin Karaca, "Entwicklung eines Lasermoduls für einen Fluoreszenztomographen," *Bachelorarbeit*, 2011.
- [7] www.thorlabs.com. [Online]. <http://www.thorlabs.com/catalogpages/V21/249.PDF>
- [9] lasercomponents.com. ADL-66701HL. [Online]. <http://www.lasercomponents.com/de/>
- [10] Conrad Electronic. [Online]. <http://www.conrad.at/ce/de/product/209893/Hitec-Standard-Servo-HS-311-Lager-Getriebe-Polyamid-JR/?ref=search>
- [11] Conrad Electronic. [Online]. <http://www.conrad.at/ce/de/product/191321/SOLARZELLE-5-V81MA/?ref=search>
- [12] G. Schwarz, "Schutz und Ansteuereinrichtung für den Bildverstärker eines Fluoreszenztomographen," *Bachelorarbeit*, 2011.
- [13] Conrad Electronic. [Online]. <http://www.conrad.at/ce/de/product/720256/INVERTER-FUeR-KALT-KATHODEN-LAMPE-300-MM>
- [14] Conrad Electronic. [Online]. <http://www.conrad.at/ce/de/product/503538/HUBMAGNET-TDS-03A-12-VDC>
- [15] Conrad Electronic. [Online]. http://www.conrad.at/ce/de/product/751227/SCHLUeSSELSCHALTER-GESAMTLAeNGE-53-MM/SHOP_AREA_17383&promotionareaSearchDetail=005
- [16] Hamamatsu Photonics. [Online]. <http://sales.hamamatsu.com/index.php?id=13200862>
- [18] M. Glitzner, "Digitaler Synthesizer für die Fluoreszenztomographie," *Projekarbeit*, 2011.
- [17] DCAM. 1394 Trade Association. [Online]. www.1394ta.org

- [19] R. Zettel, "Ansteuerung eines Galvoscaners eines Fluoreszenzthomographen," *Bachelorarbeit*, 2012.
- [20] M. Toferer, "Entwicklung eines Probenisches für die Fluoreszenzthomographie," *Bachelorarbeit*, 2010.
- [21] National Instruments. [Online]. <http://www.ni.com/labview/d/>
- [22] M. Schöllauf, "Ansteuerung der Positioniereinheit für den Graz Mark1 Fluoreszenztomographen," *Projektarbeit*, 2012.
- [23] www.trinamic.com. [Online]. http://www.trinamic.com/tmc/media/Downloads/modules/TMCM-110/TMCM-110_manual.pdf
- [24] Conrad Electronic GmbH. [Online]. <http://www.conrad.de/ce/de/product/185120/UNIPOLARER-HALLSCHALTER>
- [25] libusb-1.0. (2011) libusb: Synchronous device I/O. [Online]. http://libusb.sourceforge.net/api-1.0/group__syncio.html
- [26] Qt. [Online]. <http://qt.nokia.com/products/>
- [28] www.wikipedia.org. [Online]. [http://de.wikipedia.org/wiki/Qt_\(Bibliothek\)](http://de.wikipedia.org/wiki/Qt_(Bibliothek))
- [27] Ubuntu Linux. [Online]. <http://www.ubuntu.com/>
- [29] Apache Subversion. [Online]. <http://subversion.apache.org/>
- [30] Dimitri van Heesch. Doxygen. [Online]. www.doxygen.org
- [31] C. Kapella, "Automatisierter Oberflächenscanner," *Projektarbeit*, 2010.
- [32] www.wikipedia.org. [Online]. <http://de.wikipedia.org/wiki/JavaScript>
- [33] A. Bojahr. (2007) <http://www.u-t-b.de>. [Online]. http://www.u-t-b.de/de/pdf/Risikoanalyse_nach_der_Maschinenrichtlinie.pdf
- [34] Comité Européan de Normalisation Electroetechnique, *Europäische Norm EN 60601-1 für medizinische elektrische Geräte – Allgemeine Festlegungen für die Sicherheit*. Brüssel: CENELEC, 1996.
- [35] M. Kalis, "Spektralkamera und Beleuchtungseinrichtung für diffuse Fluoreszenztomographie ," *Diplomarbeit*, 2010.
- [36] Schaffner EMV AG. (2008) www.schaffner.com. [Online]. <http://www.schaffner.com/de/downloads/file-download/file/leakage-currents-in-power-line-filters.html>

Verzeichnis der Abbildungen

Abbildung 1: Übersichtsbild des Tomographen	10
Abbildung 2: Schematischer Aufbau des Fluoreszenztomographie Systems	13
Abbildung 3: Aufbau der Abschwächer Einheit	17
Abbildung 4: Versorgung und Vorrangschaltung für den Laser	19
Abbildung 5: Aufbau des Shutter	20
Abbildung 6: Aufbau des Verriegelungsmechanismus der Tomographen-Tür	22
Abbildung 7: Sicherheitsschaltung für die Laser Freigabe und Türverriegelung	23
Abbildung 8: Aufbau des Endschalters für die Probenrotation	26
Abbildung 9: Kollaborationsdiagramm von FTCamera	32
Abbildung 10: Kollaborationsdiagramm von FTPositioning	33
Abbildung 11: Kollaborationsdiagramm von FTLaser	35
Abbildung 12: Kollaborationsdiagramm von FTShutter	38
Abbildung 13: Kollaborationsdiagramm von FTSynthesizer	40
Abbildung 14: Kollaborationsdiagramm von FTBackLight	41
Abbildung 15: Kollaborationsdiagramm von ImageViewer	44
Abbildung 16: Kollaborationsdiagramm von Watchdog. Sie hat auf Shutter, Positionierung und Laser Zugriff	45
Abbildung 17: Kollaborationsdiagramm von ScriptMachine. Durch die Verwendung sämtlicher Ressourcen in der Klasse wird in diesem Diagramm nahezu das gesamte Steuerungsprogramm ftcontrol abgebildet.	47
Abbildung 18: Kollaborationsdiagramm von Console	48
Abbildung 19: Kollaborationsdiagramm von FTLogger	48
Abbildung 20: Impedanznachbildung mit Filter nach EN 60990	53
Abbildung 21: Kapazitive Komponenten eines Netzfilter	53

Herleitung der Kalkulation für die Synchrondemodulation

We have an excitation diode, which produces a periodic signal, which may be sinusoidal or rectangular:

$$s(t) = b_e + m(t) \quad (10)$$

Where b_e is the bias and $m(t)$ is the modulation signal while the bias can take any value. The modulation can be either sinusoidal

$$m(t) = \mu_e \sin(\omega t + \psi) \quad (11)$$

or rectangular:

$$m(t) = \mu_e \left[\sin(\omega t + \psi) + \frac{1}{3} \sin(3(\omega t + \psi)) + \frac{1}{5} \sin(5(\omega t + \psi)) + \dots \right] \quad (12)$$

Whereas μ_e denotes the modulation amplitude of the excitation source. The demodulator has a shutter function:

$$d(t) = b_d + m_d(t) \quad (13)$$

and again the modulation function is either

$$m_d(t) = \mu_d \sin(\omega t + \varphi) \quad (14)$$

Or rectangular:

$$m_d(t) = \mu_d \left[\sin(\omega t + \varphi) + \frac{1}{3} \sin(3(\omega t + \varphi)) + \frac{1}{5} \sin(5(\omega t + \varphi)) + \dots \right] \quad (15)$$

b_d is the bias at the photocathode and μ_d the modulation amplitude, which also includes the modulus of the sample transfer function. As we have a bias on both

signals we cannot just use the usual in-phase and quadrature demodulation in order to extract the real and imaginary parts. Instead we need three equations for identifying them:

The in phase product is for sinusoidal operation and reference phase $\psi = 0$:

$$\begin{aligned}
 p_I &= (b_e + \mu_e \sin(\omega t))(b_d + \mu_d \sin(\omega t + \varphi)) = \\
 &= b_e b_d + b_e \mu_d \sin(\omega t + \varphi) + b_d \mu_e \sin(\omega t) + \mu_e \mu_d \sin(\omega t) \sin(\omega t + \varphi) = \\
 &= b_e b_d + \mu_d b_e \sin(\omega t) + \mu_e b_d \sin(\omega t + \varphi) + \mu_e \mu_d \frac{1}{2} \cos(\varphi) + \mu_e \mu_d \frac{1}{2} \cos(2\omega t + \varphi)
 \end{aligned} \tag{16}$$

The terms with high frequency (all sinusoids) are averaged out and only the DC components remain, giving the I-component, i. e.:

$$I = b_e b_d + \frac{1}{2} \cos(\varphi) \tag{17}$$

In analogy the Q-component is obtained by multiplying with the cosine:

$$\begin{aligned}
 p_Q &= (b_e + \mu_e \cos(\omega t))(b_d + \mu_d \sin(\omega t + \varphi)) = \\
 &= b_e b_d + b_e \mu_d \sin(\omega t + \varphi) + b_d \mu_e \cos(\omega t) - \frac{1}{2} \mu_e \mu_d \sin(\varphi) + \frac{1}{2} \mu_e \mu_d \sin(2\omega t + \varphi)
 \end{aligned}$$

After averaging we obtain:

$$Q = b_e b_d - \frac{1}{2} \mu_e \mu_d \sin(\varphi) \tag{18}$$

We now need to get rid of the bias product $b_e b_d$ and can do so with a third equation

$$\begin{aligned}
 p_Q &= (b_e + \mu_e \cos(\omega t + \pi))(b_d + \mu_d \sin(\omega t + \varphi)) = \\
 &= b_e b_d - b_e \mu_d \sin(\omega t + \varphi) + b_d \mu_e \cos(\omega t + \pi) + \frac{1}{2} \mu_e \mu_d \sin(\varphi) - \frac{1}{2} \mu_e \mu_d \sin(2\omega t + \varphi)
 \end{aligned}$$

After averaging we obtain:

$$Q_- = b_e b_d + \frac{1}{2} \mu_e \mu_d \sin(\varphi) \tag{19}$$

We have now 3 equations with 3 unknowns and can solve. (18) and (19) give:

$$b_e b_d = \frac{Q + Q_-}{2} \quad (20)$$

Re-inserting yields:

$$\begin{aligned} \operatorname{Re}(s) &= \frac{2}{\mu_e \mu_d} \left[1 - \frac{Q + Q_-}{2} \right] = \frac{2I - (Q + Q_-)}{\mu_e \mu_d} \\ \operatorname{Im}(s) &= \frac{2}{\mu_e \mu_d} \left[Q - \frac{Q + Q_-}{2} \right] = \frac{Q_- - Q}{\mu_e \mu_d} \end{aligned} \quad (21)$$

Bedienungshandbuch

Die folgenden Seiten beinhalten das Bedienungshandbuch für die Betriebssoftware *ftcontrol*. Sie wurde so in diese Arbeit eingebettet, dass sie als eigenständiges Dokument gedruckt werden kann.



Bedienungshandbuch

Fluoreszenztomograph
Steuerungssoftware

ftcontrol

Version der Steuerungssoftware zum Zeitpunkt
der Erstellung dieses Handbuches:

ftcontrol Rev 1.5.05

Verfasser: Markus Schöllauf, TU-Graz

Erstelldatum: 26.02.12 18:08

Inhaltsverzeichnis

1	BESCHREIBUNG DER SOFTWARE	73
2	ENTWURFSMERKMALE	73
3	INBETRIEBNAHME	74
3.1	VORAUSSETZUNGEN	74
4	BEDIENUNG	74
4.1	PROGRAMMSTART	74
4.2	PROGRAMMÜBERSICHT	76
4.3	MODULE / KOMPONENTEN	79
4.3.1	<i>Kamera Modul (Setup → Camera)</i>	79
4.3.2	<i>Positionierungs-Modul (Setup → Positioning)</i>	85
4.3.3	<i>Laser Modul (Setup → Laser)</i>	89
4.3.4	<i>Galvo Modul (Setup → Galvo)</i>	92
4.3.5	<i>Shutter Modul (Setup → Shutter)</i>	93
4.3.6	<i>Synthesizer Modul (Setup → Synthesizer)</i>	95
4.3.7	<i>Hintergrundbeleuchtung Modul (Setup → Backlight)</i>	96
4.4	MESSUNGEN	96
4.4.1	<i>I/Q Messung</i>	96
4.4.2	<i>Erfassung des Phasenbildes</i>	97
4.4.3	<i>Ausführen eines Protokolls</i>	97
4.5	AUSWERTUNG	97
4.5.1	<i>Image Viewer (Run → Image Viewer)</i>	98
4.6	SKRIPTFUNKTIONEN	98
4.6.1	<i>Kamera Modul</i>	99
4.6.2	<i>Positioning – Modul</i>	102
4.6.3	<i>Laser Modul</i>	103
4.6.4	<i>Galvo Modul</i>	105
4.6.5	<i>Shutter Modul</i>	107
4.6.6	<i>Synthesizer Modul</i>	108
4.6.7	<i>Backlight Modul (Hintergrundbeleuchtung)</i>	109
4.6.8	<i>System Modul</i>	111
4.6.9	<i>Acquisition Modul</i>	112
4.6.10	<i>Ausgabe-Funktion (print)</i>	115
5	AUFBAU DES FTIMAGE BILDFORMATS	116
6	SKRIPT-BEISPIELE	118
6.1	KAMERABENUTZUNG	118
6.2	LASERBENUTZUNG	118

6.3	I-Q ₊ -Q-MESSUNG	119
6.4	DATEIOPERATIONEN	119
7	INDEX	120

1 Beschreibung der Software

ftcontrol ist eine Software zur Bedienung des Eigenbau-Fluoreszenz-Tomographen am Institut für Medizintechnik an der Technischen Universität Graz. Sie wird unter dem Betriebssystem Linux betrieben.

Mit Hilfe von *ftcontrol* kann sowohl eine Konfiguration der Hardware-Komponenten des Tomographen als auch der eigentliche Messablauf durchgeführt werden.

2 Entwurfsmerkmale

Beim Entwurf von *ftcontrol* wurde auf Modularität besonderer Wert gelegt, um einen eventuellen Ausbau der Funktionalität der Software so einfach wie möglich zu gestalten. So wurde jede Hardware-Komponente des Tomographen auch softwaretechnisch als Komponente bzw. Modul abgebildet. Diese kann (wenn notwendig) auch einzeln – ohne Starten des Hauptprogrammes – gestartet und zur Konfiguration bzw. Steuerung der jeweiligen Hardware-Komponente benutzt werden.

Die Umsetzung eines detaillierten Messablaufes am Tomographen erfolgt über Messprotokolle. Darin wird mittels der *ECMAScript*¹ (JavaScript) kompatiblen Skriptsprache *QtScript* der detaillierte Messablauf abgebildet. Alle Hardware Komponenten des Systems können im Protokoll direkt oder über Vorverarbeitungsmechanismen angesprochen und bedient werden. Weiters besteht zu diesem Zweck die Möglichkeit des Debuggings, um schrittweise das Protokoll abzarbeiten.

¹ ECMAScript ist eine von Ecma International in den Spezifikationen ECMA-262 und ISO/IEC 16262 standardisierte Skriptsprache. Sie ist in Form von Javascript, Jscript oder ActionScript weit verbreitet. Quelle: Wikipedia adaptiert

3 Inbetriebnahme

3.1 Voraussetzungen

ftcontrol ist unter dem Betriebssystem Linux lauffähig, wobei als Entwicklungsumgebung die Linux-Distribution Ubuntu 10.04 benutzt wurde.



Die Funktion von *ftcontrol* unter einer anderen Linux-Distribution wurde nicht getestet und kann daher nicht garantiert werden. Es ist jedoch sehr wahrscheinlich, dass *ftcontrol* mit etwas bzw. eventuell ohne Adaption auch auf anderen Distributionen lauffähig ist.

Um *ftcontrol* starten zu können müssen folgende Softwarepakete installiert sein:

- libqt4-* Qt4 SDK
- libusb-1.0-0 user space USB programming library
- libdc1394 IEEE1394 Scientific Camera Interface



Es existiert eine detaillierte Dokumentation des Softwareaufbaus von *ftcontrol*. Sie liegt im HTML-Format vor und befindet sich im Source-Verzeichnis des Hauptprogramms unter

trunk/doc/html/index.html

4 Bedienung

4.1 Programmstart

Ist der Computer für die Benutzung von *ftcontrol* vorbereitet, wird das Hauptprogramm mittels Aufruf des Binaries gestartet:

`$PFAD_ZUM_ARBEITSVERZEICHNIS/control/ftcontrol`

Wird das Hauptprogramm in einem Terminal gestartet werden zusätzlich Log- und System-Meldungen im Terminal angezeigt.

Anschließend werden die Hardwarekomponenten des Tomographen für die Benutzung initialisiert.

Ist dieser Initialisierungsvorgang abgeschlossen, erscheint eine Zusammenfassung der Komponenten mit ihrem aktuellen Status (vgl. Abbildung 22).

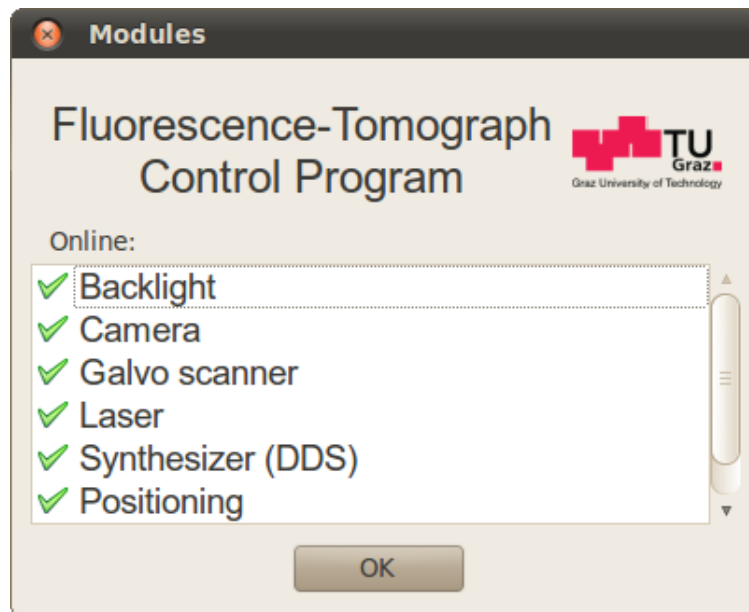


Abbildung 22: Status Übersicht aller Hardwarekomponenten des Tomographen

Abbildung 22 zeigt die Zusammenfassung. Dabei bedeutet ein ✓ Symbol auf der linken Seite jeder Komponente, dass sie ordnungsgemäß erkannt und initialisiert wurde. Wird ein ✗ Symbol angezeigt, konnte die Verbindung mit der Komponente nicht hergestellt werden oder es ist bei der Initialisierung ein Fehler aufgetreten. Dabei ist zu erwähnen, dass nicht immer alle Komponenten benötigt werden, um eine Messung durchzuführen.



Wird eine Komponente (Modul), die für die gewünschte Messung notwendig ist nicht gefunden bzw. liegt ein Fehler vor, muss nach eventueller Fehlerbehebung der Komponente das Hauptprogramm beendet und neu gestartet werden, um eine vollständige Initialisierung zu erreichen. Dieser Dialog ist auch unter **Window->Module States** aktivierbar.

Anschließend sollte das System referenziert werden. Dies wird durch eine entsprechende Meldung angezeigt. Es werden die Komponenten **Positionierung** und **Synthesizer** referenziert bzw. initialisiert. Soll dieser Vorgang nicht sofort erfolgen, kann er manuell in den entsprechenden Steuerungsdialogen (siehe Positionierungs-Modul (Setup→Positioning) und Synthesizer Modul (Setup→Synthesizer)) vorgenommen werden.

4.2 Programmübersicht

Nachdem sichergestellt ist, dass alle notwendigen Komponenten des Tomographen „online“ sind (✓), kann mit dem Programm fortgefahren werden. Abbildung 23 zeigt das Hauptfenster von *ftcontrol*.

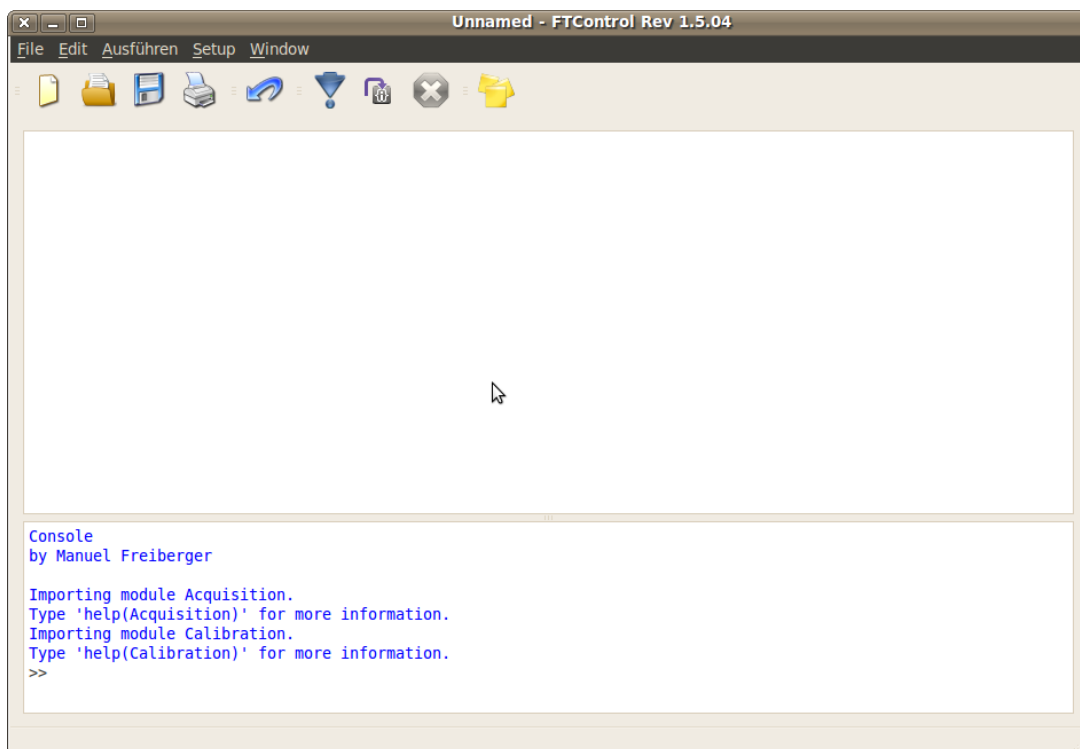
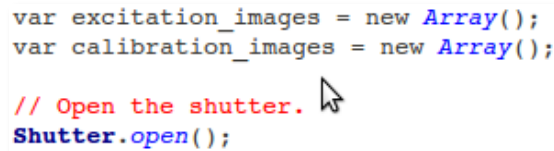
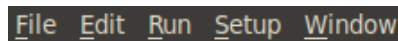


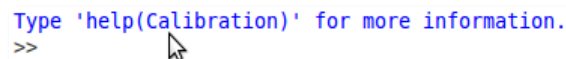
Abbildung 23: Hauptfenster von *ftcontrol*. Es ist unterteilt in einem Programmier-Bereich (oben) und einer Konsole (unten)

Das Hauptfenster enthält folgende Bereiche:

- Menü Balken:
- Aktionsbalken:
- Protokoll/Programm-Fenster:
- Konsolenfenster:



```
var excitation_images = new Array();  
var calibration_images = new Array();  
  
// Open the shutter.  
Shutter.open();
```



```
Type 'help(Calibration)' for more information.  
>>
```

Der Aktionsbalken bietet einen schnelleren Zugriff auf oft benötigte Funktionen. Diese finden sich auch in den jeweiligen Menüs im Menübalken wieder.



Es wird ein neues leeres Messprotokoll erstellt. Ein bereits geöffnetes und verändertes Protokoll wird auf Wunsch gesichert.

(File->New protocol)



Ein Messprotokoll wird geöffnet. Ein bereits geöffnetes und verändertes Protokoll wird auf Wunsch gesichert.

(File->Open protocol)



Das aktuell geöffnete Messprotokoll wird gesichert.

(File->Save protocol)



Das aktuell geöffnete Messprotokoll wird gedruckt.

(File->Print protocol)



Mithilfe dieses Links kann die letzte Operation beim Editieren des aktuell offenen Messprotokolls rückgängig gemacht werden.

(Edit->Undo)



Es kann mithilfe des Debuggers das aktuell geöffnete Messprotokoll Schritt für Schritt durchlaufen werden.

(Run->Run with debugger)



Abbruch eines aktuell laufenden Messprotokolls. Diese Funktion ist nur zu bestimmten Zeitpunkten im Protokoll möglich!



Ausführen des Messprotokolls mit vorhergehender nochmaliger Bestätigung dieser Aktion.

(Run->Run protocol)



Öffnen der Arbeitsfenster zur Steuerung des Tomographen. Es werden die Dialogfenster aller Systemkomponenten geöffnet.

(Window->Open component windows)

Die zur Ausführung und anschließender Bewertung einer Messung benötigten Funktion befinden sich unter Menüpunkt **Run**.

Menüpunkt **Run**:

Run protocol	Führt ein aktuell geöffnetes Messprotokoll aus.
Acquire phase image	Startet den Messablauf zur Erfassung eines Phasenbildes der in den Tomographen eingelegten Probe.
IQ Measurement	Führt eine Messsequenz aus, um eine Synchron-demodulation des Streulichtbildes der Probe zu erhalten. Ergebnis sind drei gemessene Bilder (I-, Q- und Q+ Anteil).
Image Viewer	Er dient zum Betrachten der gemessenen oder berechneten Probenbilder bzw. von gesicherten Bildern im proprietären FTI Format (Fluorescence Tomography Image)
Run with debugger	Führt das aktuell geöffnete Messprotokoll im Debug-Modus aus.

4.3 Module / Komponenten

Die Abbildung der gesamten Funktionalität des Tomographen auf die Steuerungsebene wird durch korrespondierende Softwareklassen bzw. Modulen erreicht. Dadurch ergibt sich eine strukturierte Planung und Ausführung des gewünschten Messablaufs. Alle Module sind unter Menüpunkt **Setup** zu finden.

Sämtliche Modul-Einstellungen werden in einer Konfigurations-Datei abgelegt und verwaltet. Diese Datei kann mithilfe der Menüpunkte **File->Export configuration** und **File->import configuration** exportiert oder importiert werden, um schnell und komfortabel große Systemänderungen vorzunehmen bzw. zu testen. Um die Standard-Konfiguration wieder herzustellen, kann die Option **Setup->Restore default configuration** benutzt werden.

4.3.1 Kamera Modul (Setup→Camera)

Dieses Modul dient der Konfiguration sowie der Bedienung der Kamera des Tomographen.

Steuerungsbereich

Bildbereich



Bildanalysebereich

ROI	Pixels	Minimum	Maximum	Mean	Standard deviation
1	5435	0	1023	647.378	321.469
3	1631	1	1023	146.292	110.529
3	5498	110	470	275.119	83.769

Abbildung 24: Kamera Steuerung

Es kann in drei Bereiche unterteilt werden:

4.3.1.1 Steuerungsbereich

Controls **Reset Camera:** Es wird ein Reset und ein anschließender Initialisierungsvorgang der Kommunikation zur Kamera sowie der Kamera selbst durchgeführt. Der aktuelle Verbindungsstatus wird durch ein Symbol rechts neben der Schaltfläche angezeigt. Dabei bedeutet ein  Symbol, dass die Kommunikation zur Kamera funktioniert. Ein  Symbol deutet auf einen Fehler in der Kommunikation mit der Kamera hin (Kabel, Stromversorgung,...)

Capture Image: Es wird ein Bild mit den aktuellen Kameraeinstellungen aufgezeichnet und im Bildbereich angezeigt.

Video Mode: Durch Drücken dieser Schaltfläche liefert die Kamera permanent Bildmaterial. Die Zeit zwischen den Bildern hängt u. a. von der eingestellten Integrationszeit ab.

Parameter **Binning:** Hier wird der Betriebsmodus der Kamera eingestellt. Er unterteilt sich bei der benutzten Kamera von Hamamatsu in den High-Speed-Mode (HS) und den High Precision-Mode (HP). Jeder der beiden Modi unterstützt die Binning-Größen 1x1, 2x2, 4x4 und 8x8 Pixel. Binning bedeutet, dass benachbarte Pixel hardwareseitig zu größeren Clustern zusammenschaltet werden. Dies reduziert dementsprechend die Größe des Bildes, erhöht jedoch dessen Lichtempfindlichkeit. Im HS-Modus arbeitet die Kamera mit einer Auflösung von 12 Bit - im HP-Modus mit 16 Bit. Im 16 Bit-Modus ist die Kamera langsamer.

Upscale Binning: Durch das Binning wird die Größe des Bildes verkleinert. Wird *Upscale Binning* angewählt und ist ein Binning Modus aktiv, wird das resultierende Bild wieder auf die originale Größe hochskaliert, d.h. das Bild hat die gleiche Pixelanzahl wie

ein Bild ohne Binning. Dabei wird ohne Glättung und unter Einhaltung der Proportionen skaliert.

Integration time: Dies ist die Integrationszeit bzw. Belichtungszeit für ein Bild. Je länger diese Zeit ist, umso mehr Licht kann auf den Bildsensor gelangen.

Gain: Hiermit kann der Verstärkungsfaktor eingestellt werden, den die Kamera hardwareseitig zur Verfügung stellt.

Filter

Durch die Filtereinstellung kann eine Art post-processing des erfassten Bildes durchgeführt werden.

Identity: Kein post-processing. Der Filter gibt die erfassten Bilddaten 1:1 - also ohne Modifikation - zur Verarbeitung weiter.

Average: Dies ist ein Moving-Average (MA) Filter, der den gleitenden Mittelwert aus n Bildern bildet und der weiteren Verarbeitung übergibt. Die Anzahl der gemittelten Bilder kann über den Parameter „Max Images“ eingestellt

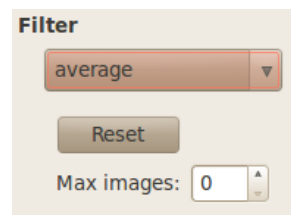


Abbildung 25: Average Filter Einstellungen

werden. Durch Drücken der Schaltfläche „Reset“ wird der Mittelwert auf null gesetzt.



Zu beachten ist, dass sich durch die akkumulierte Mittelung ein Zeitversatz vom ersten erfassten Bild bis zum gemittelten Ergebnis von n Bildern ergibt.

4.3.1.2 Bildbereich

Der Bildbereich dient zur Anzeige des erfassten Kamerabildes und zur Selektion bzw. Markierung von ROIs (Region of interest) zur numerischen Analyse im Bildanalysebereich.

Zur Analyse des Bildes stehen folgende Funktionen zur Verfügung:

- Zoom IN / OUT – Durch Verwendung der Scrollrades der Maus
- Markierung von ROIs – Durch Linksklick und Ziehen der Maus, kann ein Bereich zur numerischen Analyse markiert werden. Das Beenden der Markierung erfolgt durch linkes Doppelklicken.
Soll das Markieren abgebrochen werden, kann dies mit der Taste ESC beendet werden.

Es können mehrere ROIs markiert werden. Jede ROIs wird in die List im Bildanalysebereich eingetragen.

4.3.1.3 Bildanalysebereich

Er beinhaltet eine quantitative Auswertung aller zuvor im Bildbereich markierten ROI's. Diese Auswertung beinhaltet folgende Parameter für den jeweiligen markierten Bereich:

- Gesamtanzahl der Pixel
- Minimaler Helligkeitswert
- Maximaler Helligkeitswert
- Mittlerer Helligkeitswert
- Standardabweichung der Helligkeitswerte
- Variationskoeffizient der Helligkeitswerte



Zum Löschen einer ROI aus der Liste wird diese zunächst mit der Maus markiert und durch einen Rechtsklick und anschließendes Auswählen des Eintrags „Delete“ im Kontextmenü gelöscht.

Zur komfortablen Auswertung des Bildes besteht weiters die Möglichkeit, die Falschfarbendarstellung des erfassten Bildes zu ändern. Folgende Darstellungen stehen zur Auswahl:

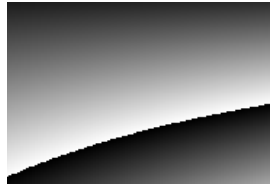


Abbildung 26: Greyscale Darstellung
Greyscale
(Graustufen)

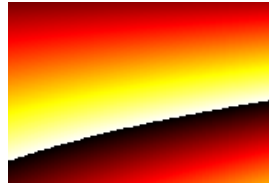


Abbildung 27: Hot Darstellung
Hot

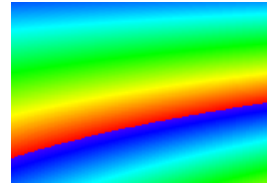


Abbildung 28: HSV Darstellung
HSV (blau-rot)

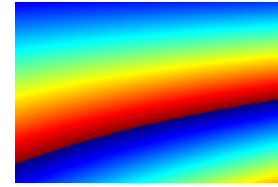


Abbildung 29: Jet Darstellung
Jet

Zur Speicherung eines Bildes wird ein proprietäres (programmeigenes) Format mit der Erweiterung „.fti“ benutzt, in dem alle Informationen bezüglich Systemeinstellung zum Zeitpunkt der Aufnahme als Metadaten gesichert werden. Der detaillierte Aufbau befindet sich im Kapitel 4.6.10. Diese Metainformation des betrachteten Bildes, werden – falls verfügbar – durch Drücken der Schaltfläche „Image Info“ angezeigt (vgl. Abbildung 30).

Acquisition Informations		
Camera Gain:	1	[-]
Integration Time:	0.020	[Sec]
Video Mode:	2x2 HS	[Mode]
Laser Attenuation:	0.3	[ODn]
DDS Amplitude:	0.000	[-]
DDS Frequency:	0.000	[MHz]
DDS Phaseshift:	0.000	[Deg]
Timestamp:	19.01.2012 17:45:16	[date]
Imagedata valid:	true	[bool]

Abbildung 30: Bild- und Erfassungs- Informationen

Die folgenden Parameter beziehen sich auf dem Zeitpunkt der Erfassung des Bildes.

Tabelle 12: Beschreibung der Metainformationen in einer FTI-Bilddatei

Parameter	Bedeutung	Einheit
Camera Gain	Eingestellter Hardware-Verstärkungsfaktor der Kamera	-
Integration Time	Integrations- bzw. Belichtungszeit der Aufnahme	Sekunden
Video Mode	Gewählter Video Modus. Er beinhaltet Auflösung und Binning.	Mode
Laser Attenuation	Gewählte Abschwächung des Lasers bei der Aufnahme	Optische Dichte (OD)
DDS-Amplitude	Amplitude des Ausgangssignals des Synthesizers	-
DDS-Frequency	Frequenz des Ausgangssignals des Synthesizers	MHz
DDS-Phaseshift	Phasenverschiebung zwischen ersten und zweiten Kanal des Synthesizers.	Grad
Timestamp	Zeitstempel des Bildes bei der Erfassung	Datum und Uhrzeit
Imagedata valid	Dieser Schalter dient internen Zwecken	bool

Weiters besteht die Möglichkeit das erfasste Bild zu sichern. Dies erfolgt durch die Schaltfläche „Save Image“.

4.3.2 Positionierungs-Modul (Setup→Positioning)

Dieses Modul steuert den Koordinatentisch sowie die Probenrotation und die Laserrotation. Es ist in einen Steuerungsteil (Abbildung 31) und einem Einstellungs-Teil (Abbildung 32) unterteilt.

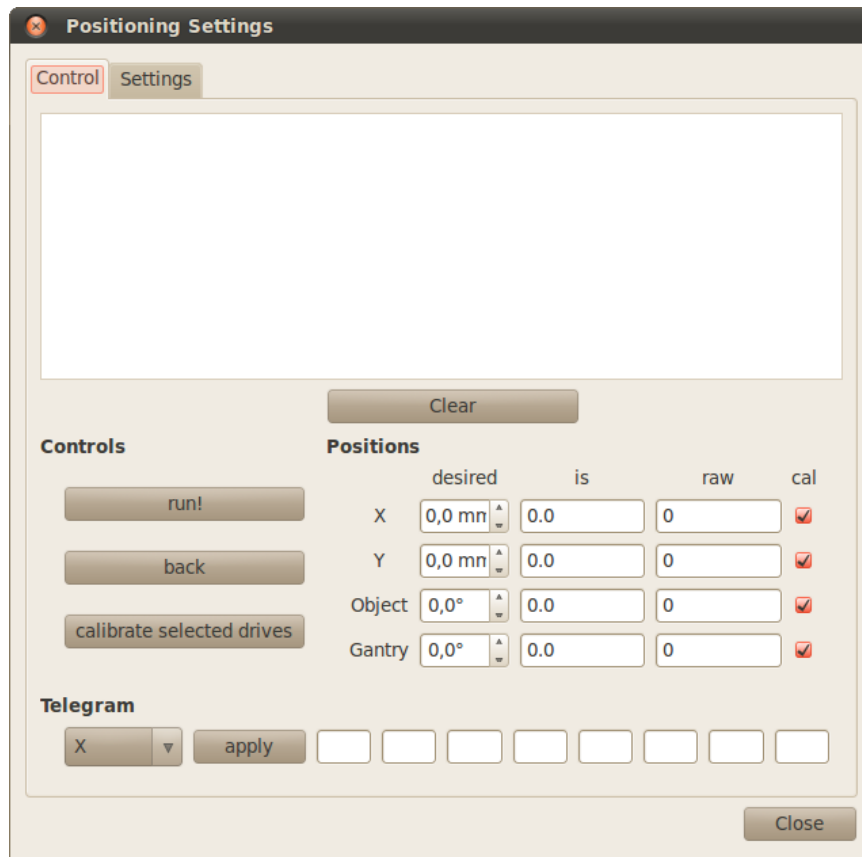


Abbildung 31: Steuerungsteil für die Proben Positionierung

Der Steuerungsteil enthält im oberen Bereich ein Log-Fenster welches Verfahren-Informationen sowie eventuelle Fehler anzeigt. Mit der Schaltfläche „**Clear**“ kann der Inhalt des Log-Fensters gelöscht werden.

Weiters beinhaltet der Steuerungsteil drei Steuerungsgruppen:

Controls **Run!:** Bewegt den Tisch (X,Y), Objektträger (Object) oder Laserarm (Gantry) auf die in der Gruppe „Positions“ eingestellte Position. Dabei wird die Position **jeder** Achse aktualisiert. Die Bewegung beginnt mit der X-Achse, Y-Achse, Objektrotation und endet mit dem Laserarm (Gantry).
Back: Bewegt **alle** Achsen, welche sich noch nicht auf Position 0.0 befinden, dorthin.

Calibrate selected drives: Startet eine Referenzfahrt mit jenen Achsen, bei welchen die Option „cal“ in der Gruppe „Positions“ aktiviert ist. Standardmäßig werden alle Achsen kalibriert. Auch hier gilt die gleiche Reihenfolge wie bei „Run!“.

Positions **Desired:** Sollposition für die Achsen X, Y, Objekt und Laserarm

Is: Ist-Position für die Achsen X, Y, Objekt und Laserarm

Raw: Schrittmotor Steps der Achsen X, Y, Objekt und Laserarm

Cal: Option für die Referenzfahrt der Achsen X, Y, Objekt und Laserarm. Sie bestimmt, welche Achse bei Betätigen der Schaltfläche „Calibrate selected drives“ referenziert werden sollen.

Telegram Diese Funktionalität ermöglicht ein Kommando-Telegramm direkt an die Schrittmotoren der Achsen zu senden. Die Checksumme des Telegramms wird automatisch beim Senden generiert und angehängt. Die Auswahl der Achse erfolgt in der linken Liste. Nach Eingabe der Telegramm-Bytes kann die Übertragung durch Betätigen der Schaltfläche „Apply“ gestartet werden.



Der detaillierte Aufbau eines Telegramms wird im Manual der Schrittmotoren beschrieben und soll hier nicht näher behandelt werden.



Achtung: Beim Bewegen des Positionier-Einrichtung besteht Verletzungsgefahr durch Quetschen! Es ist darauf zu achten, dass die

Einrichtung frei von behindernden Gegenständen bleibt, wenn sie bewegt wird.

Der Einstellungsteil (Abbildung 32) enthält die Konfigurationen aller Achsen, sowie die Einstellungen zum Positions-Logging. Die Einstellung „**Log Level**“ dient internen Zwecken wie z.B. dem Mitschnitt der Kommunikation zwischen dem Hauptprogramm und den Schrittmotoren. Die Einstellung „**Log output**“ beschreibt den Namen der Log-Datei. Wird kein absoluter Pfad angegeben, befindet sich diese Datei im Verzeichnis des Hauptprogramms.

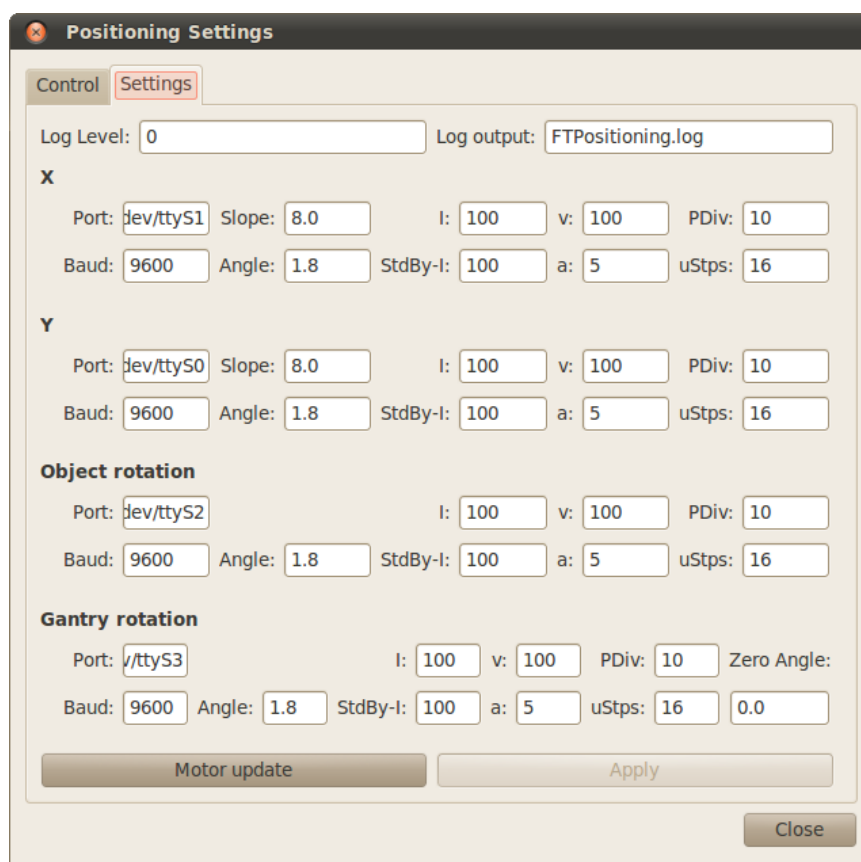


Abbildung 32: Einstellungsteil für die Proben Positionierung

Die Gruppen entsprechen den vier Achsen und beinhalten folgende Einstellungsmöglichkeiten:

Port: Der serielle COM-Port, der mit dem Schrittmotor der entsprechenden Achse verbunden ist.

Baud: Die Kommunikationsgeschwindigkeit (Baudrate) zwischen dem Hauptprogramm und dem Schrittmotor der entsprechenden Achse. Sie wird in Baud angegeben.

Slope: Die Steigung der Gewindespindel der entsprechenden Koordinatentischachse. Sie wird in mm angegeben und für die Umrechnung von Motorschritten in eine Weglänge benötigt. Diese Einstellung steht nur für die Achsen X und Y zur Verfügung da nur diese eine Spindel besitzen.

Angle: Die Winkeländerung der entsprechenden Motorachse, die durch einen Schritt des Schrittmotors verursacht wird (Schrittwinkel). Sie wird in Grad angegeben und für die Umrechnung von Motorschritten in eine Weglänge benötigt.

I: Maximaler Phasenstrom des entsprechenden Schrittmotors. Er wird in mA angegeben.

StdBy-I: Haltestrom, bei inaktiven Schrittmotor. Er wird einige Sekunden nach Motoraktivität benutzt. Er wird in mA angegeben.



Der maximale Phasenstrom wie auch der StandBy-Strom können nur in definierten Schritten angegeben werden. Detaillierte Informationen finden sich im Manual zu den Schrittmotoren.

V: Maximale Geschwindigkeit des entsprechenden Schrittmotors. Sie wird in einer proprietären Einheit angegeben. Detaillierte Informationen finden sich im Manual zu den Schrittmotoren.

A: Maximale Beschleunigung des entsprechenden Schrittmotors. Sie wird in einer proprietären Einheit angegeben. Detaillierte Informationen finden sich im Manual zu den Schrittmotoren.

Pdiv: Diese Einstellung (Pulse-Divisor) dient zur internen (motor- und hardware-bezogenen) Parameter-Berechnung und wird direkt vom Schrittmotor benutzt. Sie wird in einer proprietären Einheit angegeben. Detaillierte Informationen finden sich im Manual zu den Schrittmotoren.

uStps: Anzahl der Mikroschritte des entsprechenden Schrittmotors. Sie unterteilen einen physikalischen Schritt.



Die Einstellung der Mikroschritte kann nur in definierten Schritten erfolgen (2ⁿ). Detaillierte Informationen finden sich im Manual zu den Schrittmotoren.

Zusätzlich besitzt die Achse des Laserarmes (Gantry rotation) die Einstellung „**Zero-Angle**“. Sie wird benutzt, um nach erfolgreicher Referenz-Fahrt der Achse den eingestellten Wert anzunehmen (statt 0.0).

Werden Einstellungen verändert, kann die neue Motorkonfiguration zu Programm-Laufzeit an die Schrittmotoren gesendet werden. Dies geschieht durch Betätigen der Schaltfläche „**Apply**“. Sie ist nur dann aktiv, wenn eine Konfigurationsänderung vorliegt. Weiter wird nur jener Schrittmotor aktualisiert, dessen Konfiguration sich geändert hat.

Um die gesamte Konfiguration für **alle** Achsen neu zu senden - unabhängig ob Einstellungen geändert wurden – kann die Schaltfläche „**Motor update**“ benutzt werden.

4.3.3 Laser Modul (Setup→Laser)

Dieses Modul steuert den Laser sowie die Abschwächer-Einheit (Filterrad) des Tomographen.



Sicherheitshinweis: Der Tomographen besitzt eine Laserquelle der Klasse 3b! Es wurden diverse Sicherheitsmechanismen in Hardware und Software implementiert, um größtmögliche Sicherheit beim Betrieb des Tomographen zu bieten. Es ist jedoch weiterhin darauf zu achten, alle erforderlichen Sicherheitsaspekte zu beachten.



Achtung! Das Lasermodul kann per Software-Einstellungen aktiviert bzw. deaktiviert werden. Zusätzlich besteht die Möglichkeit über einen Schlüsselschalter einen Einricht- bzw. Wartungs-Modus zu aktivieren. In diesem Wartungsmodus wird die Sicherheitsabschaltung der Laserstrahlung beim Öffnen der Tomographen-Tür deaktiviert! Dadurch kann ungehindert Laserstrahlung austreten!

Eine Bedienung in diesem Modus ist daher ausschließlich befugten Personen gestattet.

Das Modul besteht aus einem Steuerungsteil (Abbildung 33) und einem Einstellungsteil (Abbildung 34).

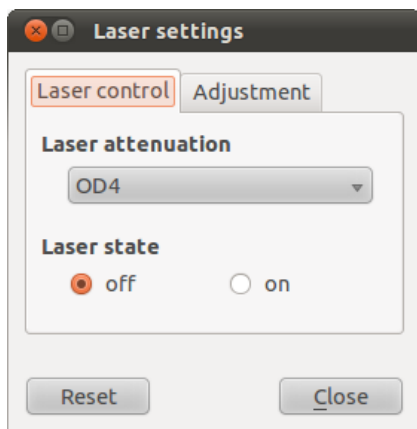


Abbildung 33: Steuerung des Lasers und der Abschwächer-Einheit / Filtrerrad



Abbildung 34: Justierung der Abschwächer-Einheit / Filtrerrad

Der Steuerungsteil beinhaltet folgende Elemente:

Laser attenuation: Die Laser Abschwächung dient der Reduzierung der Intensität des Lasers in insgesamt 6 Stufen (optische Dichten):

- | | |
|-----------------------|--------|
| 1. keine Abschwächung | 4. OD2 |
| 2. OD0.3 | 5. OD3 |
| 3. OD1 | 6. OD4 |

Laser state: Aktivierung bzw. Deaktivierung der Laserstrahlung. Zur tatsächlichen Aktivierung ist eine weitere Quittierung erforderlich



Achtung! Durch Änderung dieses Schalters kann es zur Emission von **Laserstrahlung** der Klasse 3b kommen!

Die Schaltfläche **Reset** dient zur Deaktivierung des Lasers und zur Rücksetzung der Laser-Abschwächung auf die höchste und sicherste Stufe (OD4).

Zur Justierung der Abschwächer-Einheit (Abbildung 34) werden folgende Elemente benutzt:

Justieren: Wird dieser Schalter aktiviert, werden das Eingabefeld und der Positions-Schieber freigegeben.

Eingabefeld: Es beinhaltet den aktuellen Positionswert (in einer proprietären Timingeinheit) der aktuell im Steuerungsteil eingestellten Abschwächer-Stufe. Dieser Wert repräsentiert die Position des Filtrerrades und des darauf befindlichen Filters. Er ist so zu wählen, dass der – der Abschwächer-Stufe entsprechende - Filter möglichst zentrisch im Laserstrahl zu liegen kommt.

Positions-Schieber: Durch Verschieben kann die Position des Filtrerrades kontinuierlich verändert werden. Dabei entsprechen die Schieber Endlagen (links und rechts) der maximalen Rotation (mechanische Endlagen) des Filtrerrades.

Apply: Nach beendeter Justierung der aktuellen Abschwächer-Stufe kann die Einstellung durch Betätigen dieser Schaltfläche an die Laser-Komponente des Tomographen gesendet und nicht-flüchtig gespeichert werden. Dadurch kann das Filtrerrad auch manuell mit den Tasten an der Komponente bedient werden (ohne Hauptprogramm bzw. Kommunikation).



Achtung! Für diese Justierung ist es notwendig, im Wartungsmodus (Schlüsselschalter) zu arbeiten. Es ist darauf zu achten, dass die geltenden Sicherheitsvorkehrungen eingehalten werden (**geeignete** Schutz- bzw. Justierbrille, Signalisierung für andere Personen, Absicherung des Gefahrenbereichs, ...).

Um die Sicherheit zu erhöhen, besitzt die Laser-Komponente einen Heartbeat-Mechanismus, der im Fehlerfall (z.B. bei Absturz des Steuerungsrechners,...) den Laser sicher abschaltet. Weiters wird der Laser bei Start und Beendigung des Hauptprogramms automatisch abgeschaltet.

4.3.4 Galvo Modul (Setup→Galvo)

Das Galvo-Spiegelsystem dient zur exakten Ausrichtung und Positionierung des Laserstrahls während der Messung. Es ermöglicht eine Auslenkung in X- und Y-Richtung und wird über die Galvo Steuerung bedient (Abbildung 35, Abbildung 36 und Abbildung 37).



Sicherheitshinweis: Bei der Verwendung des Galvo-Systems kann es zur unkontrollierten Laserstrahlungs-Exposition kommen!

Das Galvo Modul besteht aus drei Gruppen:

Position: Durch Bewegen des Fadenkreuzes mit der Maus wird der Laserstrahl positioniert. Eine numerische Positionierung ermöglichen die Eingabefelder für Azimut (X-Achse) und Elevation (Y-Achse).

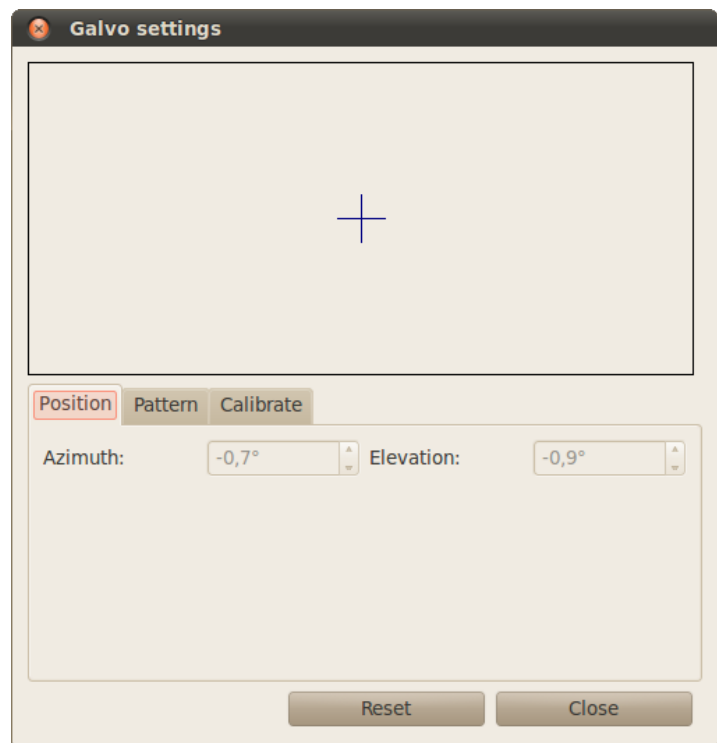


Abbildung 35: Positionierung des Galvo-Systems

Pattern: Es ist möglich, mit dem Laserstrahl ein Muster mithilfe des Galvo-Systems zu erzeugen. Die Wiederholgeschwindigkeit variiert je nach gewählten Muster.

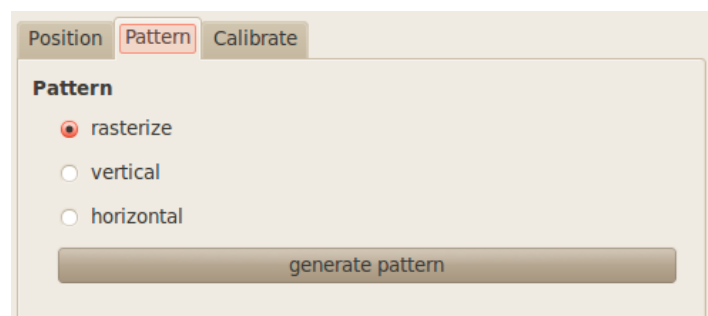


Abbildung 36: Mustererzeugung mit dem Galvo-System

Es stehen drei mögliche Muster zur Verfügung:

- **Rasterize** – Dabei wird der Laserstrahl zeilenweise im gesamten Aussteuerbereich des Galvo-Systems horizontal **und** vertikal ausgelenkt.
- **Vertical** – Ausgehend von der vorher definierten Position des Lasers wird eine vertikale Auslenkung durchgeführt. Die Auslenkung erfolgt bis an die Aussteuergrenzen.
- **Horizontal** - Ausgehend von der vorher definierten Position des Lasers wird eine horizontale Auslenkung durchgeführt. Die Auslenkung erfolgt bis an die Aussteuergrenzen.

Durch Betätigen der Schaltfläche **generate pattern** wird das gewählte Muster erzeugt.

Calibrate: Damit die Positionierung des Laserstrahles auf das zu erfassende Kamerabild bezogen werden kann, besteht die Möglichkeit einer Kalibrierung.

Abbildung 37: Kalibrierung des Galvo-Systems

4.3.5 Shutter Modul (Setup→Shutter)

Die Shutter-Komponente dient dazu, den Bildverstärker vor zu viel Umgebungslicht bzw. zu hoher Laserstrahlung zu schützen. Dazu wird ein Verschlussmechanismus (Shutter) benutzt. Er besitzt eine Solarzelle zur Messung der Helligkeit und wird für die Erstellung der Negativlisten (siehe Negativlistenenerstellung) verwendet.



Abbildung 38: Excess Light

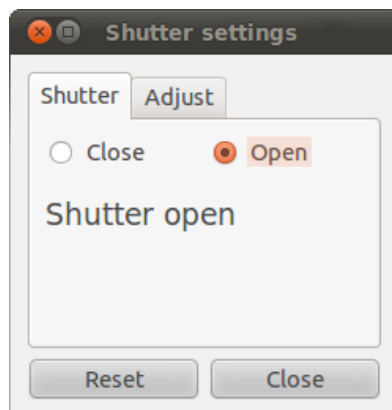


Abbildung 39: Shutter Steuerung mit Status

Die Steuerung erfolgt durch Betätigen der Schalter **Open** und **Close** (Abbildung 39). Dies ist jedoch nur möglich, wenn keine Gefahr für den Bildverstärker

besteht. Überschreitet das Umgebungslicht eine vorgegebene Helligkeit, erscheint in der Statusanzeige eine entsprechende Information (siehe Abbildung 38). Sie wird auch benutzt, um den aktuellen Zustand des Shutter anzuzeigen. Zusätzlich wird der Shutter beim Starten und Beenden des Hauptprogramms zwecks Sicherheit geschlossen.



Abbildung 40: Shutter Position Justierung

Um den Shutter bzw. die Verschlussblende möglichst exakt vor den Bildverstärker ausrichten zu können, kann seine Position im Bereich **Adjust** eingestellt werden. Durch Aktivieren des Schalters **adjust** kann im Eingabefeld die Position (in der Einheit eines proprietären Timingwertes) direkt oder über den Schieberegler eingestellt werden (Abbildung 40). Durch Betätigen der Schaltfläche **Apply**

wird die Position zur Shutter-Komponente übertragen und nicht-flüchtig gespeichert.

4.3.5.1 Negativlistenenerstellung

Die Negativliste enthält Daten jener Konstellationen der Positionier-Einrichtung, die während eines Messablaufs zu einem Überschreiten der zulässigen Umgebungshelligkeit des Bildverstärkers führen würden. Zu Erstellung dieser Liste kann zwischen zwei Modi gewählt werden:

1. Erstellung der Negativliste **durch Simulation** des gesamten Messablaufs: Hier wird die Kamera deaktiviert und mit maximaler (keine Abschwächung) Laserleistung bei geschlossenem Shutter die Positionen bestimmt, an denen der Shutter **Excess Light** meldet. Diese Methode ist jedoch mehr oder weniger zeitaufwendig, bedingt durch die Komplexität des Messablaufs.
2. Erstellung der Negativ-Liste **während** des Messablaufs (on-the-fly): Vor **jeder** Aktivierung des Lasers durch den Messablauf wird der Shutter geschlossen, der Laser aktiviert und geprüft, ob eine übermäßige Strahlung vorhanden ist. Ist dies nicht der Fall, wird der Shutter wieder geöffnet und die Lasersteuerung dem Messablauf übergeben. Wird der Zustand **Excess Light** gemeldet, wird

für diese Position die Laseraktivierung unterbunden. Der Messablauf wird weitergeführt.

Die Auswahl des entsprechenden Modus erfolgt über den Schalter

Setup->General->On-The-Fly Excess Light Check.

4.3.6 Synthesizer Modul (Setup→Synthesizer)

Zur Erzeugung der Modulationsfrequenz und der Phasenverschiebung für die Messungen wird ein DDS-Synthesizer benutzt. Die Steuerung erfolgt mittel Synthesizer Modul (Abbildung 41).

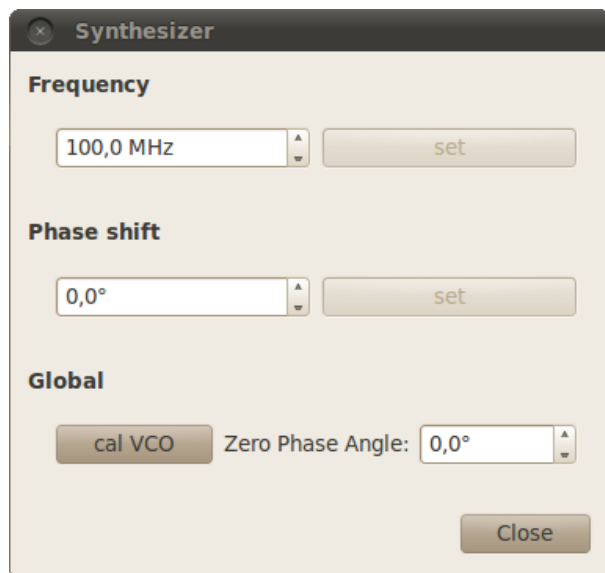


Abbildung 41: Synthesizer Steuerung

Die Steuerung besitzt drei Bereiche:

Frequency – Die für die Messung zu erzeugende Frequenz kann im Eingabefeld eingestellt werden. Bei Veränderung des Wertes wird automatisch die neue Frequenz generiert. Zusätzlich besteht die Möglichkeit über die Schaltfläche **set** die Frequenzsynthese zu starten.

Phase shift – Der Synthesizer besitzt zwei Kanäle, welche die gleiche

Frequenz ausgeben. Die Phasenlage zwischen den beiden Kanälen kann im Eingabefeld eingestellt werden. Bei Veränderung des Wertes wird automatisch die neue Phasenlage verwendet. Zusätzlich besteht die Möglichkeit über die Schaltfläche **set** die Phasenverschiebung zu erzeugen.

Global - Um den DDS zu kalibrieren bzw. hardwarebezogene Initialisierungen vorzunehmen, wird die Schaltfläche **cal VCO** benutzt. Diese Funktion wird auch bei Starten des Hauptprogramms ausgeführt. Da bei Initialisierung des DDS beide Kanäle eine willkürliche bzw. zufällige Phasenlage besitzen besteht die Möglichkeit, einen temporären Phasenabgleich mittels der Einstellung **Zero Phase Angle** vorzunehmen. Dieser Abgleich bleibt jedoch nur solange bestehen, bis eine erneute

Initialisierung des DDS erfolgt (z.B. Zurücksetzen des DDS, Aus/Einschalten, cal VCO, ...).

4.3.7 Hintergrundbeleuchtung Modul (Setup→Backlight)

Die Hardware der Probenbeleuchtung wird auch für die Steuerung des Shutter benutzt. Demnach stellt dieses Modul eine Ausnahme in Hinblick auf eine eindeutige Zuordnung von Hardware-Komponenten zu Software-Modulen dar. Die Steuerung der Beleuchtung zeigt Abbildung 42. Mithilfe des Schalters **State** kann die Beleuchtung ein- bzw. ausgeschaltet werden. Die Helligkeit wird mit dem



Abbildung 42: Beleuchtungs-Steuerung

Schieberegler **Brightness** eingestellt. Da es sich bei der Hintergrundbeleuchtung um eine Kaltkathodenröhre handelt, wird eine Mindestzündspannung des Inverters benötigt, um die Röhre zu zünden. Um jedoch weiterhin eine Helligkeitseinstellung zu ermöglichen, kann der kleinste noch mögliche Helligkeitspegel mithilfe des Schiebereglers **operation threshold** eingestellt werden.

Die Aktivierung des Schalters **Smart Ignition** bewirkt bei Einschalten der Beleuchtung, für die Dauer einer halben Sekunde die volle Helligkeit (Inverter Spannung) bevor auf die eingestellte Helligkeit heruntergedimmt wird. So wird sichergestellt, dass auch bei geringem Helligkeitspegel die

Beleuchtung ordnungsgemäß funktioniert. Durch Betätigen des Schaltfläche **Reset** wird die Beleuchtung ausgeschaltet.

4.4 Messungen

4.4.1 I/Q Messung

Eine Aufgabe des Tomographen ist es, ein Phasenbild von an einer Probe gestreutem Licht zu erfassen. Dazu wird der Mechanismus eines Synchrondemodulators benutzt. Durch Messen des durch die Probe hindurch durchtretenden bzw. gestreuten modulierten Laserlichts in den Phasenlagen 0° , 90°

und -90° relativ zu einer willkürlichen Nullphase können drei Bilder erfasst werden. Sie entsprechen der I- und Q- bzw. \bar{Q} -Komponente eines Synchronmodulators.



Der gesamte Ablauf dieser Messung wird durch Betätigen des Schalters **Run->IQ Measurement** ausgeführt. Die Ergebnisbilder werden temporär im Verzeichnis des Hauptprogramms abgelegt:

- I_IMAGE.FTI
- Q_IMAGE.FTI
- QM_IMAGE.FTI

4.4.2 Erfassung des Phasenbildes

Das Phasenbild kann aus der I/Q Messung errechnet werden. Dazu wird zuerst eine I/Q Messung durchgeführt und anschließend die Berechnung von Imaginär- und Realteil als je ein Bild mit der Bezeichnung **REAL_IMAGE.FTI** und **IMAG_IMAGE.FTI** im Verzeichnis des Hauptprogramms abgelegt. Daraus wird das Phasenbild mit dem Namen **PHASE_IMAGE.FTI** erstellt.

4.4.3 Ausführen eines Protokolls

Ein Protokoll enthält den gesamten Messablauf und kann durch Betätigen der Schaltflächen **Run->Run Protocol**  oder innerhalb des Protokoll-Debuggers mittels **Run->Run with debugger**  gestartet werden. Eine Funktions-Referenz für die Erstellung von Messprotokollen findet sich in Kapitel Skriptfunktionen

4.5 Auswertung

Je nach Messablauf bzw. Protokoll fallen die erfassten Daten in Form von Bilddateien an. Diese können zur Betrachtung bzw. zur weiteren Auswertung mit Hilfe des integrierten Image Viewers geöffnet werden.

4.5.1 Image Viewer (Run->Image Viewer)

Der Viewer (Abbildung 43) enthält zur Darstellung und Analyse der Bildinhalte einen Bildbereich und Bildanalysebereich der den Elementen des Kamera-Moduls (Bildbereich) gleicht und auch in diesem Kapitel detailliert behandelt wird. Zusätzlich besteht die Möglichkeit der Sicherung und das Öffnen von FTI-Bilder durch Betätigen der Schaltflächen **Save** und **Open**.

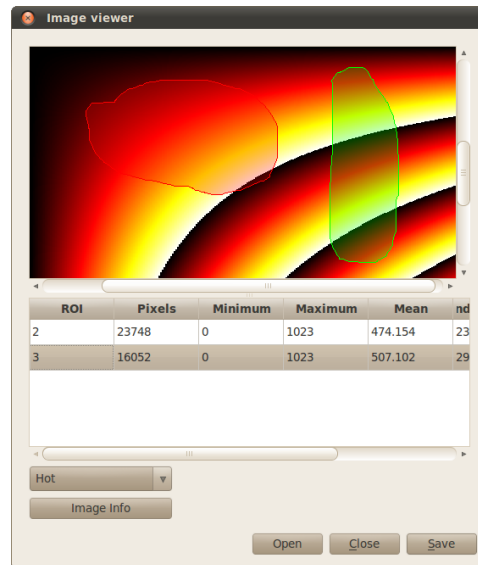


Abbildung 43: Image Viewer zur Bildauswertung

4.6 Skriptfunktionen

Die Erstellung von Messprotokollen wird mithilfe der Skriptsprache **JavaScript** realisiert. Durch Benutzung ihrer mächtigen Funktionalitäten in Kombination mit den integrierten Modul-Funktionen kann ein komplexer Messablauf geschaffen werden. Alle Module können über das Scripting mittels bereitgestellter Methoden angesprochen und auf diese zugegriffen werden. Im Folgenden werden diese Methoden im Überblick beschrieben.

Alle Methoden der jeweiligen Module können durch Aufruf von

Modulname.Methode(Parameter,...)

bzw. für Eigenschaften

Modulname.Eigenschaft = Parameter

benutzt werden. Dabei gibt es einige Methoden, die Übergabe- und/oder Rückgabeparameter besitzen.

Für weitere Informationen stehen einige Messprotokolle bzw. Beispiele im Verzeichnis des Hauptprogramms unter **control/protocols** zur Verfügung.

4.6.1 Kamera Modul

Modulname für Scripting: **Camera**

<p>void acquireImage ():</p> <p>Erfasst ein Kamerabild, wartet jedoch nicht bis das Bild fertig aufgenommen wurde.</p>
<p>void acquireImageBlocking ()</p> <p>Erfasst ein Kamerabild und wartet bis es fertig aufgenommen wurde.</p>
<p>void close ()</p> <p>Beendet die Kommunikation mit der Kamera.</p>
<p>bool open ()</p> <p>Startet bzw. öffnet die Kommunikation mit der Kamera.</p>
<p>void reset ()</p> <p>Setzt die Kommunikation mit der Kamera bzw. die Kamera selbst zurück.</p>
<p>void setGain (int gain)</p> <p>Setzt die Hardware-Verstärkung der Kamera auf <i>gain</i>.</p>
<p>void setIntegrationTime (double seconds)</p> <p>Setzt die Integrationszeit (Belichtungszeit) auf <i>seconds</i>.</p>
<p>void setVideoMode (int mode)</p> <p>Setzt den zu benutzenden Video Modus auf <i>mode</i> (siehe Verfügbare Video Modes).</p>
<p>void unsetError ()</p> <p>Setzt einen Fehler der Kamera zurück.</p>

void setWatchdogLock (bool locked)
Bei <i>locked</i> = true wird die Aufnahmefunktion der Kamera deaktiviert und bei Starten einer Bilderfassung sofort ein Dummy-Bild zurückgegeben.
FTImage currentImage ()
Gibt das zuletzt erfasste Kamerabild zurück.
int error () const
Gibt aktuelle Fehlernummer der Kamera zurück.
double getIntegrationTime ()
Gibt aktuell eingestellte Integrationszeit (Belichtungszeit) in Sekunden zurück.
int getVideoMode ()
Gibt aktuell eingestellten Video Modus zurück (siehe Verfügbare Video Modes).
bool hasError ()
Gibt <i>true</i> zurück, wenn die letzte Erfassung fehlerhaft war.
bool isAcquiring ()
Gibt <i>true</i> zurück, wenn gerade eine Erfassung erfolgt.
QString error
Gibt die aktuelle Fehlerbezeichnung der Kamera zurück.
int gain
Setzt die Hardware Verstärkung im Bereich [0,255], oder gibt diese zurück.
double integrationTime
Setzt die Integrationszeit (Belichtungszeit) in Sekunden oder gibt diese zurück.

int **mode**

Setzt den Video Modus oder gibt diesen zurück (siehe **Verfügbare Video Modes**).

bool **open**

Öffnet die Kommunikation zur Kamera oder gibt ihren Status zurück.

Verfügbare Video Modes:

HIGH_SPEED_1x1 = 0

HIGH_SPEED_2x2 = 1

HIGH_SPEED_4x4 = 2

HIGH_SPEED_8x8 = 3

HIGH_PRECISION_1x1 = 4

HIGH_PRECISION_2x2 = 5

HIGH_PRECISION_4x4 = 6

HIGH_PRECISION_8x8 = 7

Diese Bezeichnungen sind wie folgt zu benutzen:

Modulname.metaObject.Bezeichnung

Beispiel:

Camera.mode = Camera.metaObject.HIGH_PRECISION_1x1

4.6.2 Positioning – Modul

Modulname für Scripting: **Positioning**

<p>bool calibrate ()</p> <p>Startet eine Referenzfahrt auf allen Achsen und gibt <i>true</i> zurück wenn kein Fehler aufgetreten ist.</p>
<p>bool calibrate (CalibrationFlags flags)</p> <p>Startet eine Referenzfahrt auf den durch <i>flags</i> bestimmten Achsen und gibt <i>true</i> zurück wenn kein Fehler aufgetreten ist. <i>flags</i> wird durch Oderverknüpfung definiert.</p>
<p>void reset ()</p> <p>Setzt die Kommunikation zu den Achsen zurück.</p>
<p>void setGantryRotation (double angle)</p> <p>Setzt die Position des Laserarms auf <i>angle</i> in Grad.</p>
<p>void setObjectRotation (double angle)</p> <p>Setzt die Position des Objektträgers auf <i>angle</i> in Grad.</p>
<p>void setX (double x)</p> <p>Setzt die Position der X-Achse auf <i>x</i> in mm.</p>
<p>void setY (double x)</p> <p>Setzt die Position der Y-Achse auf <i>y</i> in mm.</p>
<p>double getGantryRotation ()</p> <p>Gibt die aktuelle Rotation des Laserarms in Grad zurück.</p>
<p>double getObjectRotation ()</p> <p>Gibt die aktuelle Rotation des Objektträgers in Grad zurück.</p>

double **getX** ()

Gibt die aktuelle Position der X-Achse in mm zurück.

double **getY** ()

Gibt die aktuelle Position der Y-Achse in mm zurück.

QByteArray **setCommand** (int drive, QByteArray cmd)

Erlaubt die direkte Ausführung eine Kommandos (*cmd*) auf der angegebenen Achse *drive*. Nähere Informationen finden sich im Handbuch der Schrittmotoren.

4.6.3 Laser Modul

Modulname für Scripting: **Laser**

void **reset** ()

Setzt die Kommunikation zum Laser zurück.

void **setEnabled** (bool enable)

Aktiviert bzw. deaktiviert den Laser. **Vorsicht!** Laserstrahlung!

void **setLaserAttenuation** (int attenuation)

Setzt die Laser-Abschwächung auf *attenuation* (**siehe Verfügbare Laser-Abschwächungen**).

void **setServo** (unsigned int val)

Setzt die Position des Filtrrad-Servos mithilfe des Wertes *val*. (Ein interner Timingwert).

unsigned int **getServo** ()

Gibt den internen Timingwert der aktuellen Position des Filtrrad-Servos zurück.

void **writeServoVal** (unsigned int idx)

Schreibt den aktuell eingestellten Timingwert in den nichtflüchtigen Speicher der Laser-Komponente und verknüpft ihn mit der Positionsnummer *idx*.

bool **hasError** ()

Gibt *true* zurück, bei einem Fehler des Lasers.

<p>bool isSafetyEnabled () Gibt <i>true</i> zurück, wenn die Sicherheitsfreigabe vom systeminternen Watchdog erteilt wurde.</p>
<p>int attenuation Setzt die Laser-Abschwächung (siehe Verfügbare Laser-Abschwächungen) oder gibt diese zurück.</p>
<p>bool enable Aktiviert bzw. deaktiviert den Laser oder gibt diesen Status zurück. Vorsicht! Laserstrahlung!</p>
<p>string error Gibt die letzte anliegende Fehlerbeschreibung zurück.</p>
<p>bool open Öffnet bzw. Schließt die Kommunikation zum Laser oder gibt diesen Status zurück.</p>
<p>bool state Gibt den Status des Lasers zurück (ein/aus).</p>

Verfügbare Laser-Abschwächungen:

<p>None = 0</p> <p>OD0_3 = 1</p> <p>OD1 = 2</p> <p>OD2 = 3</p> <p>OD3 = 4</p> <p>OD4 = 5</p>
--

Diese Bezeichnungen sind wie folgt zu benutzen:

Modulname.metaObject.Bezeichnung

Beispiel:

Laser.attenuation = Laser.metaObject.OD4

4.6.4 Galvo Modul

Modulname für Scripting: **Galvo**

<p>void reset ()</p> <p>Setzt die Kommunikation zur Galvo-Komponente zurück</p>
<p>void setAzimuth (double azimuth_angle)</p> <p>Setzt den Azimutwinkel (X-Achse) des Galvosystems auf <i>azimuth_angle</i> (in Grad).</p>
<p>void setAzimuthAngleToInternalFactor (double factor)</p> <p>Setzt den Faktor für die interne Umrechnung zwischen Grad und internen Einheiten.</p>
<p>void setAzimuthOffset (int x)</p> <p>Setzt den Offset Faktor x (in internen Einheiten) zur Umrechnung zwischen Grad und internen Einheiten.</p>
<p>void setElevation (double elevation_angle)</p> <p>Setzt den Elevationswinkel (Y-Achse) des Galvosystems auf <i>elevation_angle</i> (in Grad).</p>
<p>void setElevationAngleToInternalFactor (double factor)</p> <p>Setzt den Faktor für die interne Umrechnung zwischen Grad und internen Einheiten.</p>
<p>void setElevationOffset (int y)</p> <p>Setzt den Offset y (in internen Einheiten) zur Umrechnung zwischen Grad und internen Einheiten .</p>
<p>void setPosition (QPoint pos)</p> <p>Setzt die Position des Galvo-Systems auf <i>pos</i> mithilfe eines 2D-Punktes.</p>

void **setPosition** (int x, int y)

Setzt die Position des Galvo-Systems auf x und y in internen Einheiten.

void **setX** (int x)

Setzt die X-Koordinate des Galvo-Systems auf x.

void **setY** (int y)

Setzt die Y-Koordinate des Galvo-Systems auf y.

bool **hasError** () const

Gibt *true* zurück bei einem Galvo-Fehler.

double **azimuth**

Setzt den Azimuth-Winkel (X-Ablenkung) in Grad oder gibt diesen zurück.

double **elevation**

Setzt den Elevations-Winkel (Y-Ablenkung) in Grad oder gibt diesen zurück.

string **error**

Gibt die Beschreibung des letzten Fehlers zurück.

bool **open**

Öffnet die Kommunikation zum Galvo-System oder gibt diesen Status zurück.

QPoint **position**

Setzt die Position des Galvo-Systems mithilfe eines Punktes oder gibt diese zurück.

Die Umrechnung zwischen Winkel und internen Einheiten erfolgt durch folgende Beziehungen:

$$x = k_x * \varphi + d_x \quad (2)$$

$$y = k_y * \theta + d_y \quad (3)$$

Die Parameter haben folgende Bedeutung:

x, y die internen DAC-Werte für Azimut und Elevation, die zum Galvo-Modul übertragen werden

φ, θ die Winkel für Azimut und Elevation in Grad

k_x, k_y die Skalierungsfaktoren für Azimut und Elevation (*setAzimuthAngleToInternalFactor()*, *setElevationAngleToInternalFactor()*)

d_x, d_y die Offsetwerte für Azimut und Elevation (*setAzimuthOffset()*, *setElevationOffset()*)

4.6.5 Shutter Modul

Modulname für Scripting: **Shutter**

void **close** ()

Schließt den Shutter.

void **open** ()

Öffnet den Shutter.

void **reset** ()

Setzt die Kommunikation zum Shutter zurück.

void **setShutterOpen** (bool open)

Öffnet oder schließt den Shutter wenn *open true* oder *false* ist.

int **shutterState** ()

Gibt den aktuellen Shutter Zustand zurück.

bool **isClosed** ()

Gibt *true* zurück, wenn der shutter geschlossen ist.

QString **error**

Gibt die Fehlerbeschreibung des letzten Fehlers zurück.

bool **interfaceOpen**

Gibt den Zustand der Kommunikation zum Shutter zurück.

4.6.6 Synthesizer Modul

Modulname für Scripting: **Synthesizer**

void **calibrateVCO** ()

Initialisiert und kalibriert die DDS-Komponente.

void **setFrequency** (double frequency)

Setzt die Frequenz beider DDS Kanäle auf *frequency* in MHz.

void **setPhaseOffset** (double offset)

Setzt den Null-Phasenwinkel des DDS auf *offset* in Grad.

void **setPhaseShift** (double phase_shift)

Setzt den Phasenwinkel zwischen den beiden DDS-Kanälen.

void **reset** ()

Setzt die Kommunikation zur DDS-Komponente zurück .

double **frequency** ()

Gibt die aktuell generierte Frequenz des DDS zurück.

double **phase** ()

Gibt den aktuellen Phasenwinkel zwischen beiden DDS-Kanälen zurück .

double **phaseOffset** ()

Gibt den Null-Phasenwinkel zwischen den beiden Kanälen des DDS zurück.

string **errorString** ()

Gibt die Fehlerbeschreibung des letzten Fehlers zurück.

bool **hasError** ()

Gibt *true* zurück wenn ein Fehler anliegt.

4.6.7 Backlight Modul (Hintergrundbeleuchtung)

Modulname für Scripting: **BackLight**

void **reset** ()

Setzt das Backlight Interface zurück.

void **setBrightness** (int percentage)

Setzt die Helligkeit der Hintergrundbeleuchtung auf *percentage* in Prozent.

void **setEnabled** (bool enable)

Aktiviert bzw. deaktiviert die Hintergrundbeleuchtung.

void **setOperationThreshold** (int threshold)

Setzt den internen Mindestwert für die PWM-Versorgungserzeugung, um noch eine Helligkeitsregelung zu gewährleisten.

void **setSmartIgnitionEnable** (bool enable)

Aktiviert bzw. deaktiviert die *Smart Ignition*-Funktion zur sicheren Helligkeitsregelung.

int **brightness**

Setzt die Helligkeit oder gibt diese zurück.

bool **enable**

Aktiviert bzw. deaktiviert die Hintergrundbeleuchtung.

string **error**

Gibt die Beschreibung des letzten Fehlers zurück.

bool **interfaceOpen**

Gibt *true* zurück, wenn das Interface zum Backlight Modul geöffnet ist.

4.6.8 System Modul

Modulname für Scripting: **System**

cd <i>directory</i>
Wechselt das Verzeichnis im Datei-System.
mkdir (<i>new_directory</i>)
Erzeugt ein neues Verzeichnis am aktuellen Ort.
bool exists (<i>directory</i>)
Gibt <i>true</i> zurück wenn das Verzeichnis <i>directory</i> existiert.
string expand (<i>path</i>)
Ersetzt alle in <i>path</i> vorkommenden Umgebungsvariablen (z.B. \$HOME) und gibt den neuen Pfad zurück.
void sleep (<i>milliseconds</i>)
Wartet <i>milliseconds</i> lang.
void resetCamera ()
Setzt die Kamera auf Standardeinstellungen zurück: Gain=0, Integration Time = 1, Video Mode = High Precision – 1x1 binning.
void resetLaser ()
Setzt den Laser auf Standardeinstellungen zurück: Enabled = <i>false</i> , Attenuation = OD4.

void **savelmages** (stack_of_images, image_name_prefix)

Sichert alle Bilder in *stack_of_images* im aktuellen Arbeitsverzeichnis unter einem Dateinamen, beginnend mit *image_name_prefix*, gefolgt von einem – dem jeweiligen Bildtyp entsprechenden – Kürzel (z.B. *_real* für ein Real-Bild, *_qp* für ein QP-Bild,...).

void **showImage**(image)

Zeigt das Bild *image* im Image Viewer.

void **uniqueDir**(directory_prefix)

Erzeugt ein Verzeichnis, das mit *directory_prefix* beginnt und einen Zeitstempel im Namen beinhaltet. So wird sichergestellt, dass es nur das eine Verzeichnis mit diesem Namen gibt.

void **resetAll** ()

Führt *resetCamera()* und *resetLaser()* aus.

4.6.9 Acquisition Modul

Modulname für Scripting: **Acquisition**

var **acquireIqpQm**(*i_angle* = 0, *qp_angle* = *i_angle* + 90, *qm_angle* = *i_angle* - 90)

Ermittelt ein I-Qm-Qp-Bildset mit den angegebenen Parametern, und gibt es als *ftimage*-Array zurück. Werden keine Parameter angegeben, wird die entsprechende Voreinstellung benutzt. Dabei entsprechen die Parameter:

i_angle: Winkel bei dem das I-Bild ermittelt wird.

qp_angle: Winkel bei dem das Qp_Bild ermittelt wird.

qm_angle: Winkel bei dem das Qm_Bild ermittelt wird.

var **acquireIQpQmSeries**(num_imagesets,i_angle,qp_angle,qm_angle)

Erfasst einen Satz mit *num_imagesets* I-Qp-Qm-Bildsets und gibt diesen (zweidimensionales Array) zurück. Dabei entsprechen die Parameter jenen von acquireIQpQm().

var **acquireIQpQmAvg**(averages)

Ermittelt den Mittelwert über *averages* I-Qp-Qm-Bildsets und gibt diesen wieder als Bildset zurück.

var **acquireMagnitudePhase**(i_angle, qp_angle, qm_angle)

Ermittelt ein Bildset von Betrag und Phase mit den angegebenen Parametern und gibt dieses zurück. Dabei entsprechen die Parameter jenen von **acquireIQpQm()**.

Autoscaling

Stellt Funktionen zur Bilderfassung mit optimalen Belichtungszeiten zur Verfügung (siehe detaillierte Beschreibung).

Autoscaling members:

minTargetValue

Setzt die untere Grenze für den Ziel-Dynamikbereich oder gibt diesen zurück.

maxTargetValue

Setzt die obere Grenze für den Ziel-Dynamikbereich oder gibt diesen zurück.

minIntegrationTime

Setzt die minimale Integrationszeit als Abbruchkriterium für die Belichtungsoptimierung oder gibt diese zurück.

maxIntegrationTime
Setzt die maximale Integrationszeit als Abbruchkriterium für die Belichtungsoptimierung oder gibt diese zurück.
maxNumImages
Setzt die maximale Anzahl von Testbildern als Abbruchkriterium für die Belichtungsoptimierung oder gibt diese zurück.
numLargestPixels
Setzt die Anzahl von Pixel für die Berechnung des Median Wertes zur Belichtungsoptimierung oder gibt diese zurück.
Offset
Setzt den Dunkelwert für die Belichtungsoptimierung oder gibt diesen zurück.
var acquireImage()
Erfasst ein Bild mit angepasster Belichtungszeit und gibt dieses zurück.
setDarkImage(image)
Stellt <i>offset</i> auf den dunkelsten Bildwert im angegebenen Bild <i>image</i> zur weiteren Belichtungsoptimierung ein.
setTargetRangeFraction(lower_fraction, upper_fraction)
Berechnet und setzt den Ziel-Dynamikbereich mithilfe von <i>lower_fraction</i> und <i>upper_fraction</i> in einem – auf den Kamera-Dynamikbereich normierten Intervall [0,1]. Dabei ist zu beachten, dass <i>upper_fraction</i> > <i>lower_fraction</i> .

```
var acquireIQmQp(i_angle = 0, qp_angle = i_angle + 90, qm_angle = i_angle - 90)
```

Ermittelt ein I-Qm-Qp-Bildset mit den angegebenen Parametern und angepasster Belichtungszeit und gibt es als fimage-Array zurück. Werden keine Parameter angegeben, wird die entsprechende Voreinstellung benutzt. Dabei entsprechen die Parameter:

i_angle: Winkel bei dem das I-Bild ermittelt wird.

qp_angle: Winkel bei dem das Qp_Bild ermittelt wird.

qm_angle: Winkel bei dem das Qm_Bild ermittelt wird.

4.6.10 Ausgabe-Funktion (print)

Zur Ausgabe von Informationen oder Fehlermeldungen innerhalb eines Messprotokolls kann die Funktion **void print(string)** werden.

5 Aufbau des FTImage Bildformats

Das Bildformat **FTImage**, das von *ftcontrol* benutzt wird, verwendet die Auszeichnungssprache XML² zum Sichern der Bilddaten sowie zum Einbetten der Metainformationen (vgl. Abbildung 44).

```
<?xml version="1.0" encoding="UTF-8"?>
<ftimage>
  <parameters invalid="0" timestamp="Mi. Feb 1 01:05:19 2012">
    <position gantryAngle="45.000" objectAngle="10.000" x="20.000" y="40.000"/>
    <camera gain="1" integrationTime="0.020" mode="1"/>
    <laser attenuation="1"/>
    <dds frequency="0.000" phase="0.000"/>
  </parameters>
  <data height="512" width="512">AAAAA...xbDa</data>
</ftimage>
```

Abbildung 44: Aufbau des FTI-Bildformates

Das gesamte Bild samt Metainformationen liegt in einem **<ftimage>** tag. Innerhalb dieses tags befinden sich die beiden tags **<parameters>** und **<data>**.

<parameters> hat folgende Attribute:

- **invalid:** Dieser Schalter dient zur Markierung des Bildes als gültig bzw. für interne Zwecke.
- **timestamp:** Zeitstempel zur Protokollierung des Zeitpunktes der Bilderzeugung.

<data> enthält die Base64 kodierten Bildinformationen und hat folgende Attribute:

- **height:** Höhe des Bildes in Pixel
- **width:** Breite des Bildes in Pixel

² XML (Extensible Markup Language) ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten. Quelle: Wikipedia adaptiert.

Alle Metainformationen sind als eigene Einträge verfügbar und werden in den **<parameters>** Tag eingebettet:

<position> hat folgende Attribute:

- **gantryAngle:** Winkel des Laserarms zum Zeitpunkt der Bilderfassung in Grad
- **objectAngle:** Winkel des Objektträgers zum Zeitpunkt der Bilderfassung in Grad
- **x:** Position der X-Achse des Verfahr-Tisches zum Zeitpunkt der Bilderfassung in mm
- **y:** Position der Y-Achse des Verfahr-Tisches zum Zeitpunkt der Bilderfassung in mm

<camera> hat folgende Attribute:

- **gain:** Hardware-Verstärkung der Kamera zum Zeitpunkt der Bilderfassung
- **integrationTime:** Integrationszeit zum Zeitpunkt der Bilderfassung in Sekunden
- **mode:** Video Modus der Kamera zum Zeitpunkt der Bilderfassung (siehe 4.6.1 Kamera Modul, Verfügbare Video Modi)

<laser> hat folgende Attribute:

- **attenuation:** Laser Abschwächung zum Zeitpunkt der Bilderfassung (siehe 4.6.3 Laser Modul, Verfügbare Laser-Abschwächungen).

<dds> hat folgende Attribute:

- **frequency:** Synthesizer Frequenz zum Zeitpunkt der Bilderfassung in MHz
- **phase:** Phasenlage der Synthesizer Kanäle zum Zeitpunkt der Bilderfassung in Grad

6 Skript-Beispiele

6.1 Kamerabenzutzung

Das folgende Beispielskript setzt die Integrationszeit der Kamera auf 0.5 Sekunden und benutzt den *High-Precision*-Modus mit einem 2x2 Binning. Danach wird der Shutter geöffnert und ein Bild erfasst.

```
// First reset the camera
System.resetCamera();

// Set an integration time of 0.5 seconds.
Camera.integrationTime = 0.5;

// Change the mode to high-precision with 2x2 binning.
Camera.metaObject.HIGH_PRECISION_2x2;

// before an image can be acquired the shutter has to be
opened

Shutter.open();

// Then tell the camera to take an image and store it

// Acquire an image.
Camera.acquireImageBlocking();

// Store the last image.
var image = Camera.currentImage();

// For maximum safety, close the shutter afterwards.
Shutter.close();
```

6.2 Laserbenutzung

Folgende Zeilen stellen die Abschwächung des Lasers auf OD0.3 und aktivieren den Laser.

```
Laser.attenuation = Laser.metaObject.OD0_3;

Laser.enable = true;
```

6.3 I-Q+-Q.-Messung

Um einen Stapel von I-Q+-Q.-Sets zu erfassen, kann wie folgt vorgegangen werden.

```
Acquisition.acquireIQpQm();
```

Um eine Messung zu starten, bei der alle drei Komponenten um 45° verschoben sind, wird folgendes Kommando verwendet:

```
Acquisition.acquireIQpQm(45);
```

6.4 Dateioperationen

Dieses Beispiel legt ein neues Verzeichnis an und wechselt in dieses bzw. benutzt es als Arbeitsverzeichnis. Weiters gibt es Informationstext aus.

```
// Suggest a new directory under ${HOME}/mymeasurements/  
var dir = System.uniqueDir("${HOME}/mymeasurements/");  
  
// Create this directory and change into it.  
System.mkdir(dir);  
System.cd(dir);  
  
// Some information for the user.  
print("Saving data to directory: " + dir);
```

7 Index

- Ableitkapazitäten, 53
- Ableitstrommessung, 52, 60
- Abschwächer, 89, 90, 91
- Average, 81
- Azimut, 25
- Belichtungszeit, 80, 84, 99, 100, 114, 115
- Berührungsstrommessung, 52, 60
- Betriebssoftware, 61
- Bildanalysebereich, 79, 82, 98
- Bildbereich, 79, 80, 82, 98
- Bildverstärker, 24
- Binning, 80, 84
- Capture, 80
- configuration, 79
- Control-Transfer-Modus*, 27
- DCAM, 24
- DDS, 24, 84, 95, 108, 109
- DNA, 9
- Elevation, 25
- EN 50178, 60
- EN 60601, 49
- EN 60990, 52
- Endschalter, 26
- Excess Light, 38, 94, 95
- Falschfarbendarstellung, 83
- Farbstoff, 9
- Filterrad, 11, 17
- Filterscheiben, 17
- Firmware, 20
- Fluorophor, 9
- Fotokathode, 20
- FTBackLight*, 41
- FTCamera*, 31
- FTGalvo*, 36
- FTImage, 45
- FTImageView*, 43
- FTLaser*, 34
- FTPositioning*, 33
- FTShutter*, 38
- FTSynthesizer*, 40
- Gain, 80, 84, 111
- Galvo, 25, 92, 93, 105, 106
- Gantry, 86, 89
- Hallschalter, 26
- Hardware, 62
- Hauptfenster, 76, 77
- High-Precision Readout, 24
- High-Speed Readout, 24
- Hintergrundbeleuchtung, 12, 21, 96, 109, 110
- Hochfrequenz, 9
- Hubmagnet, 22
- Identity, 81
- Image Viewer, 43
- Impedanznachbildung, 52
- Integrationszeit, 80, 99, 100, 113, 114
- Inverter, 21
- JavaScript, 73, 98
- Justieren, 91
- Kaltkathodenröhre, 21
- Kamera, 11, 24, 79, 80, 84, 94, 98, 99, 100, 101, 111, 114, 117
- Kommunikationsschnittstellen, 15
- Komponenten, 13
- Konsole, 48

Koordinatentisch, 25
 Laser, 18
 Lasermodul, 18
LaserSafetyHeartbeat, 35
 Lebensdauermessung, 9
 Leckstrom, 52
 Log, 74, 85, 87
 Logger, 48
 Maschinenrichtlinie, 49
 MDF, 52
 Motorkonfiguration, 89
 Negativ-Liste, 94
 Negativliste, 38
 Netzfilter, 53
 Normalbetrieb, 23
 OD
 Optische Dichte, 84
 optische Dichte, 11
PANdrive, 25
 Positioniereinheit, 9
 Positionierung, 25
 Pulsweitenmodulation, 21
QScriptEngine, 46
 Referenzierung, 26
 Risikoanalyse, 49, 55
 Run, 77, 78, 86, 97, 98
 Schlüsselschalter, 23
 Schrittmotor, 86, 87, 88, 89
 Schutzleiterstrom, 52
 Schutzmechanismus, 12
 Script Engine, 46
ScriptMachine, 46
 Servo, 17, 20
 Shutter, 12, 20, 93, 94, 96, 107, 108
 Sicherheitskomponenten, 12, 21
 Sicherheitsschaltung, 23
 Signalaufbereitung, 20
 Signal-Slot, 30
 Software, 27
 Spiegelsystem, 25, 92
 Spindel, 88
 Steuerungsbereich, 79, 80
 Synchrondemodulation, 42
 Synthesizer, 24, 40, 76, 95, 108
 Systemkonfiguration, 49
 Türkontakt, 22
 Türverriegelung, 21
 Übersicht, 13
 Upscale, 80
 USB
 Universal serial bus, 28
 USB-Modul
 Aufbau, 27
 Video Mode, 80, 84, 111
 Wartungsbetrieb, 23
 Wartungs-Modus, 89
 Watchdog, 44
 Zero-Angle, 89
 Zündspannung, 21