

Verfahrsteuerung und Stromauslese am QCLAM-Spektrometer

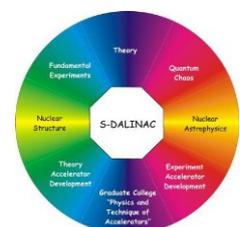
Drive control and current measurement at the QCLAM-Spectrometer
Bachelor-Thesis von Gerhart Steinhilber
Dezember 2013



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Physik
Institut für Kernphysik

Gefördert durch die DFG im Rahmen des SFB 634.



Verfahrsteuerung und Stromauslese am QCLAM-Spektrometer
Drive control and current measurement at the QCLAM-Spectrometer

Vorgelegte Bachelor-Thesis von Gerhart Steinhilber

1. Gutachten: Prof. Dr. Dr. h.c. Norbert Pietralla
2. Gutachten: Dipl.-Phys. Simela Aslanidou

Tag der Einreichung:

Zusammenfassung

Die vorliegende Bachelor-Thesis besteht aus zwei Teilen.

Der erste Teil besteht aus der Entwicklung einer neuen Verfahrsteuerung für das QCLAM-Spektrometer. Durch die neue Steuerung soll es möglich sein, Streuwinkel mit hoher Genauigkeit, reproduzierbar anzufahren. Weiterhin soll es die Möglichkeit geben, einen eingestellten Winkel automatisch anfahren zu lassen. Bei der Entwicklung der neuen Steuerung wurde auf eine hohe Betriebssicherheit geachtet. Das reproduzierbare Anfahren eines Winkels mit hoher Genauigkeit wird durch einen Drehgeber realisiert, der an einem Rad des QCLAM-Spektrometers angebracht ist, und mit einer speicherprogrammierbaren Steuerung umgesetzt. Der neue Steuerkasten konnte wegen Vakuumtests und des anschließenden Strahlbetriebs zeitlich nicht im Rahmen dieser Bachelor-Arbeit am QCLAM-Spektrometer getestet werden. Die Tests mit einem Tischaufbau verliefen erfolgreich.

Der zweite Teil der Arbeit beschäftigt sich mit der Weiterentwicklung der Stromauslese am Faraday-Cup hinter dem QCLAM-Spektrometer. Die Stromauslese soll sowohl im Kontrollraum für die Strahloptimierung bereit stehen, als auch in die Messdatenaufnahme des QCLAM-Spektrometers eingebunden werden. Um dies umsetzen zu können, wurde die Stromauslese am Faraday-Cup hinter dem Lintott-Spektrometer überarbeitet und erweitert. Dadurch war es direkt möglich, den gemessenen Strahlstrom im Kontrollraum anzeigen zu lassen. Erste Tests während der Strahlzeit Herbst 2013 verliefen erfolgreich.

Für das Speichern des Strahlstroms mit den Messdaten des QCLAM-Spektrometers, wurde der frequenzgewandelte Strahlstrom von einem Scaler registriert. Der Scaler befindet sich in einem Crate, welches Teil der Datenaufnahme ist. Erste Messungen mit einem Testsignal verliefen erfolgreich.



Abstract

This Bachelor Thesis is divided into two parts.

In the first part a new drive control for the QCLAM-Spectrometer is presented. Through this new drive control it should be possible to drive the spectrometer to scattering angles with a high accuracy. Besides it should also be possible to reach a given scattering angle automatically. In the development great value was set in high operating safety. To guarantee the needed high accuracy a rotary encoder was installed at one wheel of the QCLAM-Spectrometer. This rotary encoder is connected with a Programmable Logic Controller. Due to vacuum tests and scheduled beam time at the S-DALINAC it was not possible to perform tests of the drive control on the final setup. But all executed dry run tests have been successful.

The second part deals with a further development of the current measurement at the Faraday-Cup behind the QCLAM-Spectrometer. It was the aim to provide the measured current to the accelerator operator in the control room for beam optimization as well as in the data acquisition room to save the beam current with the data from experiments. To realize this, the existing current measurement from the Lintott-Spectrometer was extended to use it as well for the QCLAM-Spectrometer. Thus it was possible to display the measured beam current in the control room. The setup was tested during beam time in autumn 2013 with success.

To save the measured beam current with the data from experiments it was necessary to transform the measured beam current into a rectangular signal. The frequency of the rectangular signal is proportional to the beam currents absolute value. A Scaler, which is included into the data acquisition, registers the rectangular signals. Results from first measurements with a test signal will be presented.



Inhalt

1.....	Einleitung.....	1
1.1.	Elektronenstreuung.....	1
1.2.	Koinzidenzexperimente.....	5
1.3.	S-DALINAC.....	5
1.4.	Das QCLAM-Spektrometer.....	6
1.4.1.	Aufbau des Spektrometers.....	6
1.4.2.	Detektorsystem.....	8
2.....	Entwicklung einer neuen Verfahrsteuerung.....	11
2.1.	Verwendete Komponenten.....	12
2.2.	Speicherprogrammierbare Steuerung.....	13
2.3.	Steuerkasten.....	14
2.4.	Funktionsweise des Steuerungsprogramms und Bedienung.....	14
2.5.	Anzeigen auf dem Display.....	15
3.....	Strom- und Ladungsauslese am Faraday-Cup.....	17
3.1.	QM07-Elektronik und Aufbau.....	17
3.2.	Datenauslese der QM07-Elektronik mit EPICS.....	18
3.3.	Datenaufnahme und Aufbau.....	21
3.3.1.	Aufbau der Datenaufnahme.....	21
3.3.2.	Datenaufnahme und Testmessung.....	23
3.4.	Unsicherheiten der Strom- und Ladungsmessung.....	25
4.....	Fazit und Ausblick.....	27
5.....	Anhang.....	29
5.1.	easySoft 6.....	29
5.1.1.	Einleitung.....	29
5.1.2.	Bedienoberflächen.....	29
5.2.	Beschreibung der verwendeten Schaltplanelemente.....	32
5.3.	Steuerprogramm.....	34
5.4.	Bedienungsanleitung.....	41
5.4.1.	Der automatische Modus.....	41
5.4.2.	Der manuelle Modus.....	41
5.5.	Verwendete Teile.....	42
5.6.	Stromauslese mit EPICS.....	44
5.7.	Datenaufnahme des Strahlstroms mit MBS.....	51
5.7.1.	f_user.c.....	51
5.7.2.	sis_3820.h.....	55
5.7.3.	sis_3820.c.....	57
5.7.4.	Makefile.....	59



1. Einleitung

Am supraleitenden Darmstädter Elektronen-Linearbeschleuniger S-DALINAC [1] werden Experimente zur Untersuchung der Kernstruktur durchgeführt. Es stehen verschiedene Experimentierplätze zur Verfügung, an denen Photonen- bzw. Elektronenstreuexperimente realisiert werden können.

In dieser Arbeit wird Bezug auf die Elektronenstreuung bei Streuwinkeln bis zu 180° und die Möglichkeit der Durchführung von Koinzidenzexperimenten genommen.

1.1. Elektronenstreuung

Für die Untersuchung von Atomkernen ist Elektronenstreuung gut geeignet, da die elektromagnetische Wechselwirkung gut verstanden ist. Bei der Elektronenstreuung werden Elektronen an Atomkernen gestreut. Dabei ist zwischen elastischer und inelastischer Streuung zu unterscheiden. Elastische Streuung liegt vor, wenn das Elektron keine Energie an den Kern abgibt. Bei inelastischer Streuung finden Anregungen des Kerns statt. Die Energieerhaltung liefert für inelastische Streuung unter Vernachlässigung der Rückstoßenergie des Kerns

$$E_0 = E_f + E_x. \quad (1.1)$$

Hierbei sind E_0 die Energie der einfallenden Elektronen, E_f die Energie der gestreuten Elektronen und E_x die Anregungsenergie des Kerns. Der Impulsübertrag \vec{q} kann aus der Impulserhaltung gewonnen werden. Er ist gegeben durch die Differenz der Impulse \vec{p}_i und \vec{p}_f des Elektrons vor und nach der Streuung.

$$\vec{q} = \vec{p}_i - \vec{p}_f \quad (1.2)$$

Der Betrag des Impulsübertrags ist unter Vernachlässigung des Rückstoßes gegeben durch:

$$q = \frac{1}{\hbar c} \sqrt{2 \cdot E_0 (E_0 - E_x) (1 - \cos(\theta)) + E_x^2} \quad (1.3)$$

mit

$$\hbar c = 197 \text{ MeV fm.}$$

Somit ist der Impulsübertrag eine Funktion des Streuwinkels θ und der Energie E_0 der einfallenden Elektronen. Ein Maß für die Wahrscheinlichkeit der Reaktion zwischen Elektron und Kern im Raumwinkelement $d\Omega$ ist der differentielle Wirkungsquerschnitt. Der differentielle Wirkungsquerschnitt für die Streuung relativistischer Spin- $\frac{1}{2}$ Teilchen an punktförmigen Spin-0 Targets wird als Mott-Wirkungsquerschnitt $\left(\frac{d\sigma}{d\Omega}\right)_{\text{Mott}}$ bezeichnet. Er ist durch Gleichung (1.4) gegeben [2].

$$\left(\frac{d\sigma}{d\Omega}\right)_{\text{Mott}} = \frac{Z e^2 \cos^2\left(\frac{\theta}{2}\right)}{2 E_0 \sin^4\left(\frac{\theta}{2}\right)} \quad (1.4)$$

Bei der experimentellen Bestimmung des Wirkungsquerschnitts treten jedoch Abweichungen zum Mott-Wirkungsquerschnitt auf. Dies geschieht, da die Voraussetzungen für den Mott-Wirkungsquerschnitt nicht auf reale Kerne anwendbar sind. Durch die Bestimmung der Abweichungen sind Rückschlüsse auf den realen Kern möglich. Der Quotient von

experimentellem Wirkungsquerschnitt $(\frac{d\sigma}{d\Omega})_{\text{exp}}$ und Mott-Wirkungsquerschnitt wird als quadrierter Formfaktor $|F(E_0, q)|^2$ bezeichnet [2].

$$|F(E_0, q)|^2 = \frac{(\frac{d\sigma}{d\Omega})_{\text{exp}}}{(\frac{d\sigma}{d\Omega})_{\text{Mott}}} \quad (1.5)$$

Die Fouriertransformierte des Formfaktors liefert die Ladungsverteilung des Kerns. Der experimentelle Wirkungsquerschnitt kann aus gemessenen Spektren mit Gleichung (1.6) berechnet werden [2].

$$\left(\frac{d\sigma}{d\Omega}\right)_{\text{exp}} = \frac{A}{\Delta\Omega} \frac{e}{I t} \frac{M}{t_{\text{eff}} N_A} \frac{100}{\epsilon} \quad (1.6)$$

A steht für die Fläche unter dem Peak, $\Delta\Omega$ ist der Raumwinkel des Spektrometers, I steht für den Strom und t für die Zeit. Weiterhin sind M die Molare Masse des Targets, t_{eff} die effektive Dicke des Targets, N_A die Avogadrokonstante und ϵ ein Maß für die Isotopenreinheit des Targets.

Während des Streuvorgangs wechselwirkt das einfallende Elektron mit der Ladungs- und Stromdichteverteilung des Kerns durch den Austausch virtueller Photonen. In der Dirac'schen Störungstheorie erster Ordnung wird nur der Austausch eines virtuellen Photons berücksichtigt [3]. Der differentielle Wirkungsquerschnitt ist dann für den Streuwinkel θ durch

$$\left(\frac{d\sigma}{d\Omega}\right)_{\theta} = \frac{1}{4 \pi^2 (\hbar c)^2} E_i E_f \left(\frac{p_f}{p_i}\right) \left(\frac{2 J_f + 1}{2 J_i + 1}\right) f_{\text{rec}} |\langle \Psi_f | \hat{H}_{\text{int}} | \Psi_i \rangle|^2 \quad (1.7)$$

gegeben. Hierbei sind p_i und p_f die Dreierimpulse des Elektrons vor und nach der Wechselwirkung, J_i und J_f die Gesamtdrehimpulse des Kerns vor und nach der Wechselwirkung und \hat{H}_{int} der Hamilton Operator der Wechselwirkung. Die Zustände $|\Psi_i\rangle$ und $|\Psi_f\rangle$ sind die Zustände des Kerns vor bzw. nach der Streuung. Der Rückstoßfaktor f_{rec} ist durch Gleichung (1.8) gegeben [3]. M steht hier für die Masse des Kerns.

$$f_{\text{rec}} = \left(1 + \frac{2 E_i \sin^2\left(\frac{\theta}{2}\right)}{M c^2} \right)^{-1} \quad (1.8)$$

Der Hamiltonoperator \hat{H}_{int} beschreibt für Elektronenstreuung nur elektromagnetische Wechselwirkung. Er beinhaltet das Strahlungsfeld des bewegten Elektrons A^μ sowie die Ladungs- und Stromdichteverteilung j_μ des Kerns [3].

$$\hat{H}_{\text{int}} = \int j_\mu(\vec{r}, t) A^\mu(\vec{r}, t) d^3r \quad (1.9)$$

mit

$$A^\mu(\vec{r}) = (\Phi(\vec{r}), \vec{A}(\vec{r})) \quad (1.10)$$

$$j_\mu(\vec{r}) = (q(\vec{r}), \frac{1}{c} \vec{j}(\vec{r})) \quad (1.11)$$

Es ist möglich den Hamiltonoperator in eine longitudinale und eine transversale Komponente zu zerlegen. Die Coulomb-Wechselwirkung des Elektrons mit der Ladungsverteilung des Kerns wird durch die longitudinale Komponente \hat{H}_C beschrieben. Die transversale Komponente \hat{H}_T beschreibt die Wechselwirkung des Elektrons mit der Strom- und Magnetisierungsdichte des Kerns [3].

$$\hat{H}_{\text{int}} = \hat{H}_C + \vec{H}_T \quad (1.12)$$

mit

$$\hat{H}_C = \int \rho(\vec{r}) \Phi(\vec{r}) d^3r \quad (1.13)$$

$$\hat{H}_T = -\frac{1}{c} \int (\vec{j}^c(\vec{r})\vec{A}(\vec{r}) + \vec{\mu}^s(\vec{r})\vec{\nabla} \times \vec{A}(\vec{r})) d^3r \quad (1.14)$$

Hierbei sind $\rho(\vec{r})$ der Übergangsoperator der Ladungsdichte, $\vec{j}^c(\vec{r})$ der Übergangsoperator des Konvektionsstromes und $\vec{\mu}^s(\vec{r})$ der Übergangsoperator der Magnetisierungsdichte. Bei leichten Kernen ($Z\alpha \ll 1$) ist es möglich das Strahlungsfeld des virtuellen Photons als ebene Welle zu beschreiben. Hierbei ist α die Feinstrukturkonstante. Diese Näherung wird „Plane Wave Born Approximation“, oder kurz PWBA, genannt.

Für mittelschwere und schwere Kerne ist die PWBA keine sinnvolle Näherung mehr. Das Coulombfeld des Kerns verformt die Wellenfronten des gestreuten Elektrons, sodass die Annahme ebener Wellen nicht mehr zulässig ist. Bei leichten und mittleren Kernen kann diese Verformung noch durch die Einführung eines effektiven Impulsübertrags beschrieben werden. Bei schweren Kernen werden die Wellenfunktionen der Elektronen durch numerische Lösung der Dirac-Gleichung in Form einer Störungstheorie gewonnen. Diese Näherung wird als „Distorted Wave Born Approximation“ (DWBA) bezeichnet [3].

Mit der PWBA ist es möglich den Hamiltonoperator nach Multipolen zu entwickeln. Für die Multipolordnung λ bei gegebenen Gesamtdrehimpuls J und gegebener Parität π der Kernzustände folgen die Auswahlregeln für Parität und Gesamtdrehimpuls [3]. Erfüllt die Parität die Relation $\pi_i \pi_f = (-1)^\lambda$, so handelt es sich um einen elektrischen Übergang, erfüllt sie dagegen die Relation $\pi_i \pi_f = (-1)^{\lambda+1}$, handelt es sich um einen magnetischen Übergang. Die Auswahlregel für den Gesamtdrehimpuls ist durch Gleichung (1.15) gegeben.

$$|J_i - J_f| \leq \lambda \leq |J_i + J_f| \quad (1.15)$$

Durch die Multipolentwicklung, den Auswahlregeln, vernachlässigbaren Energieverlusts und dem Grenzfall relativistischer Elektronen ist es möglich, den differentiellen Wirkungsquerschnitt als eine Summe der differentiellen Wirkungsquerschnitte der Multipole $E\lambda$ und $M\lambda$ auszuschreiben. Dies ist in Gleichung (1.6) geschehen.

$$\left(\frac{d\sigma}{d\Omega}\right) = \sum_{\lambda} \left(\left(\frac{d\sigma}{d\Omega}\right)_{E\lambda} + \left(\frac{d\sigma}{d\Omega}\right)_{M\lambda} \right) \quad (1.16)$$

Für Kerne mit Gesamtdrehimpuls und Parität $J^\pi = 0^+$ sind die differentiellen Wirkungsquerschnitte durch Gleichungen (1.17) und (1.18) gegeben. Die Kinematik der Streuung wird durch die longitudinalen bzw. transversalen Faktoren V_L und V_T beschrieben [3].

$$\left(\frac{d\sigma}{d\Omega}\right)_{E\lambda} = \left(\frac{Z e^2}{E_i}\right)^2 f_{\text{rec}}(V_L |F(C\lambda, q)|^2 + V_T |F(E\lambda, q)|^2) \quad (1.17)$$

$$\left(\frac{d\sigma}{d\Omega}\right)_{M\lambda} = \left(\frac{Z e^2}{E_i}\right)^2 f_{\text{rec}}(V_T |F(E\lambda, q)|^2) \quad (1.18)$$

$$V_L = \frac{1 + \cos(\theta)}{2 (y - \cos(\theta))^2} \quad (1.19)$$

$$V_T = \frac{2y + 1 - \cos(\theta)}{4 (y - \cos(\theta))(1 - \cos(\theta))} \quad (1.20)$$

mit

$$y = 1 + \frac{E_x^2}{2 E_i (E_i - E_x)} \quad (1.21)$$

Wird die Ruhemasse des Elektrons m_0 berücksichtigt, so ergibt sich für den longitudinalen Faktor V_L folgende Form [3]:

$$V_L = \frac{\left(\frac{m_0 c^2}{E_0}\right)^2 \sin\left(\frac{\theta}{2}\right) + \cos^2\left(\frac{\theta}{2}\right)}{4 \sin^4\left(\frac{\theta}{2}\right)} \quad (1.22)$$

Für einen Streuwinkel von $\theta = 180^\circ$ ergibt sich ein ausgeprägtes Minimum. Wie in Abbildung 1 zu sehen ist, werden bei diesem Streuwinkel die elektrischen Anregungen unterdrückt, wodurch die magnetischen Übergänge dominieren. Rückwärtsstreuung ist daher für die Untersuchung magnetischer Übergänge geeignet. Experimente zur 180° -Streuung können am QCLAM-Spektrometer durchgeführt werden.

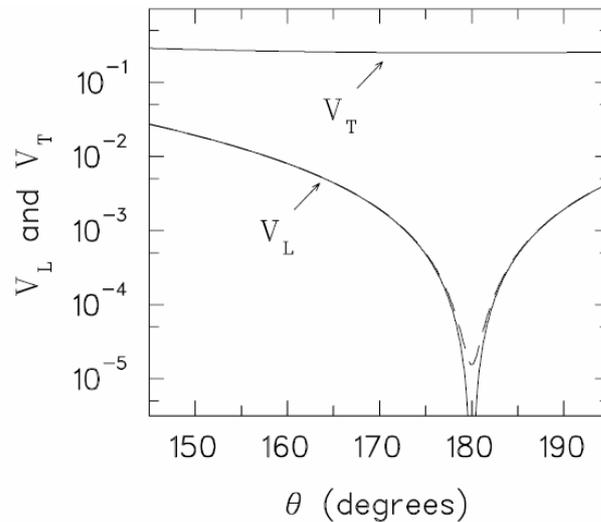


Abbildung 1: Dargestellt ist der Verlauf der Faktoren V_L und V_T für eine Einschussenergie der Elektronen von 65 MeV. Bei dem Winkel $\theta = 180$ fällt der Faktor V_L stark ab. Dadurch dominieren bei diesem Winkel die magnetischen Übergänge. [4]

1.2. Koinzidenzexperimente

Koinzidenzexperimente der Form $(e, e' x)$ haben gegenüber Einarm-Elektronenstreuexperimenten der Form (e, e') mehrere Vorteile, haben allerdings auch den Nachteil kleiner Zählraten. Die Notation $(e, e' x)$ bedeutet, dass durch die Reaktion zwischen Elektron und Kern, Teilchen vom Kern emittiert werden. Das QCLAM-Spektrometer wurde entwickelt, um solche Koinzidenzexperimente am S-DALINAC durchführen zu können.

Durch die Messung der vom Kern koinzident emittierten Teilchen, ist es möglich, den Untergrund zu unterdrücken, indem nur diejenigen Ereignisse berücksichtigt werden, bei welchen der Nachweis von gestreuten Elektronen e' , sowie Teilchen x in einem gewissen Zeitfenster erfolgte. Dies ist beispielsweise wichtig für die Messung von Riesenresonanzen. Riesenresonanzen sind kollektive Kernanregungen. In Spektren treten sie als breite Erhöhung auf. Die Messung von Riesenresonanzen wird durch einen großen, durch Bremsstrahlung verursachten, Untergrund erschwert. Der Nachweis der koinzident emittierten Teilchen ermöglicht es außerdem einen bestimmten Reaktionsvorgang zu selektieren. Zudem ist es durch Messung der Winkelverteilung der Teilchen x möglich, die Multipolarität sowie Drehimpuls und Parität des Kerns zu bestimmen [5].

Für Koinzidenzexperimente kann ein Detektor, wie beispielsweise der Siliziumball, in der Streukammer untergebracht werden [6].

1.3. S-DALINAC

Der supraleitende Darmstädter Elektronen-Linearbeschleuniger S-DALINAC wurde 1991 am Institut für Kernphysik der technischen Universität Darmstadt in Betrieb genommen [1]. Als Elektronenquelle stehen eine thermische Kathode und eine Quelle für spinpolarisierte Elektronen zur Verfügung [7]. Der maximale Strahlstrom liegt bei $60 \mu\text{A}$, solch starke Strahlströme sind aber auf den Bereich vor den Rezirkulationen begrenzt. Unter Ausnutzung der zweifachen Rezirkulation ist es möglich Elektronenenergien von bis zu 130 MeV zu erreichen. Nach Verlassen des Hauptbeschleunigers kann der Elektronenstrahl zu verschiedenen Experimentierplätzen weitergeleitet werden. Zwei dieser Experimentierplätze sind das Lintott-Spektrometer (auch 169° -Spektrometer genannt) und das QCLAM-Spektrometer. Hinter beiden Spektrometern befindet sich jeweils ein Faraday-Cup, in dem der Elektronenstrahl aufgefangen wird. Die Faraday-Cups ermöglichen es auch den Strahlstrom zu messen. In Abbildung 2 sind der Elektronenbeschleuniger und die Experimentierplätze dargestellt.

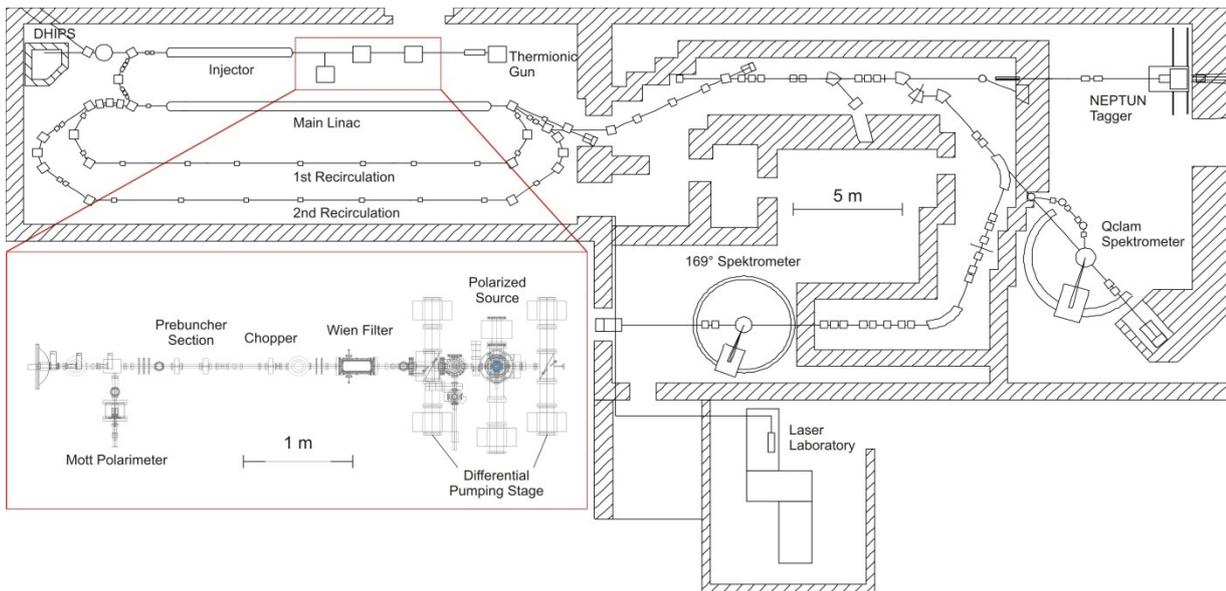


Abbildung 2: Dargestellt sind der S-DALINAC und die Experimentierplätze. Für den S-DALINAC stehen eine thermische und eine spinpolarisierte Elektronenquelle zur Verfügung. Unter Verwendung der zweifachen Rezirkulation können Elektronenenergien von maximal 130 MeV erreicht werden, welche für Experimente am NEPTUN Tagger und beiden Spektrometern benutzt werden können. In der Beschleunigerhalle Hinter beiden Spektrometern befindet sich ein Faraday-Cup, die zur Messung des Strahlstroms während Experimenten an den Spektrometern dienen. [8]

1.4. Das QCLAM-Spektrometer

Das QCLAM-Spektrometer ist eines der beiden Magnetspektrometer am S-DALINAC. In der Streukammer gestreute Elektronen werden in einem Dipolmagneten, abhängig von ihrem Impuls, auf unterschiedlichen Kreisbahnen abgelenkt. Dadurch treffen die Elektronen an unterschiedlichen Orten, mit unterschiedlichen Winkeln im Detektorsystem auf. Durch Messung der Eintrittswinkel in das Detektorsystem, sowie der Durchstoßorte der Elektronen, kann die Energie der einfallenden Elektronen bestimmt werden.

Mit dem Spektrometer können Streuwinkel im Bereich von 25° bis 155° angefahren werden. Dies ist wichtig, um bei konstanter Einschussenergie der Elektronen unterschiedliche Impulsüberträge messen zu können.

Wichtige Eigenschaften des Spektrometers sind die große relative Impulsakzeptanz von $\pm 10\%$ und die große Raumwinkelakzeptanz von 35 msr [9], wodurch eine höhere Zählrate erreicht wird. Dadurch können auch Experimente mit kleinen Wirkungsquerschnitten wie etwa Koinzidenzexperimente und 180° -Streuung durchgeführt werden.

1.4.1. Aufbau des Spektrometers

Der Name QCLAM setzt sich aus einem Q für Quadrupolmagnet und dem englischen Wort clam, welches Muschel bedeutet und die Form des Dipolmagneten beschreibt, zusammen. Elektronen, die aus dem Hauptbeschleuniger kommen, werden an einem Target in der Streukammer gestreut. Die gestreuten Elektronen können durch eine Öffnung von der Streukammer in das Spektrometer gelangen. Dort passieren sie als erstes den horizontal fokussierenden Quadrupolmagneten. Anschließend gelangen die gestreuten Elektronen in den Dipolmagneten. Durch diesen werden sie abhängig von ihrem Impuls auf unterschiedlichen Radien abgelenkt. Nach Verlassen des Dipolmagneten treten die gestreuten Elektronen durch eine Vakuumabschlussfolie, die durch ein Kevlargetebe gestützt wird, aus dem Vakuum aus. Hinter der Vakuumaustrittsfolie befindet sich das Detektorsystem. Es besteht aus drei

Vieldrahtdriftkammern, einem Szintillator und einem Čerenkovzähler. Die abgelenkten Elektronen durchqueren zuerst die drei Vieldrahtdriftkammern, dann den Szintillator und zum Schluss den Čerenkovzähler. In Abbildung 3 ist ein Querschnitt durch Spektrometer, Streukammer und Detektorsystem zu sehen.

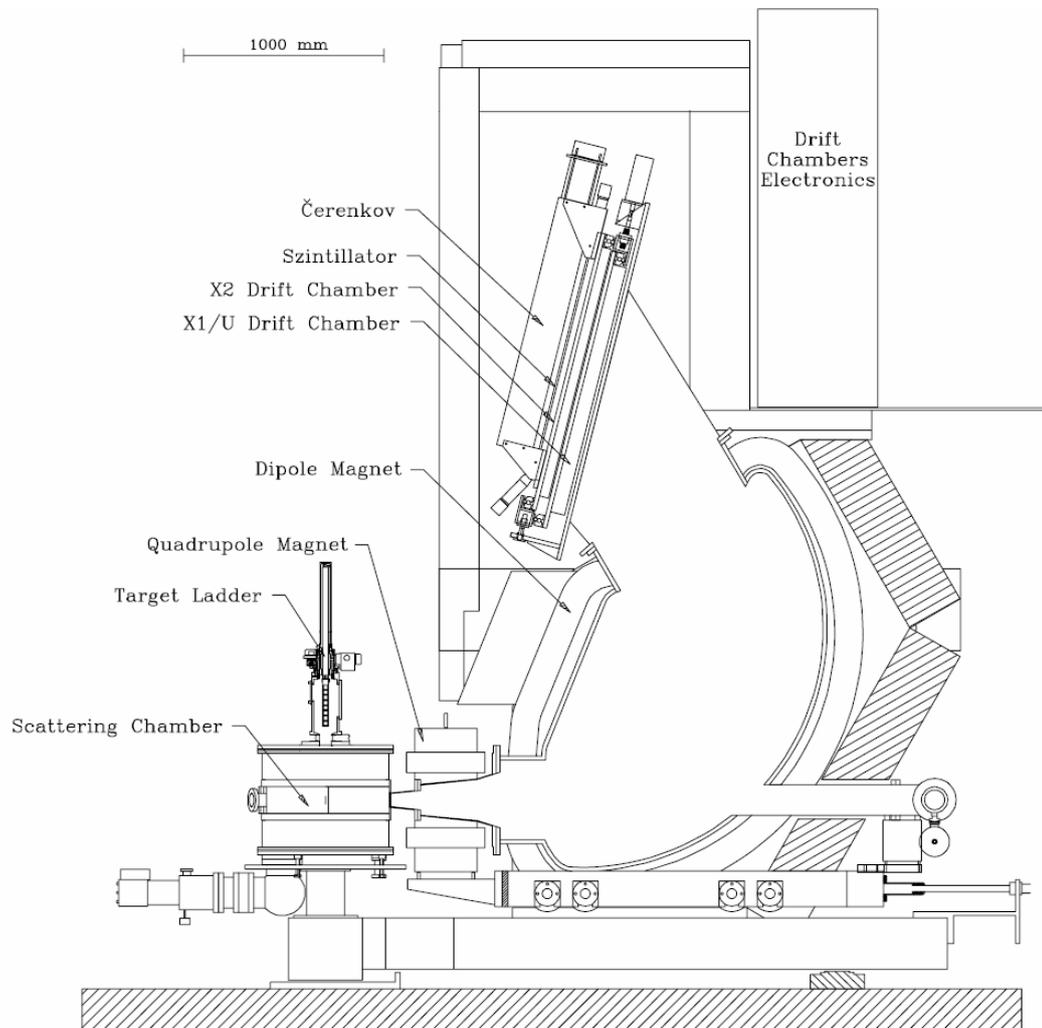


Abbildung 3: Dargestellt ist ein Querschnitt durch die Symmetrieebene des QCLAM-Spektrometers. Zu sehen sind die Streukammer, das Spektrometer und das Detektorsystem, welches in einer Abschirmung aus Polyethylen und Blei untergebracht ist. [4]

Eine Besonderheit am Institut für Kernphysik der TU Darmstadt ist, dass 180° -Streuexperimente durchgeführt werden können. Für die Durchführung von 180° -Streuung wird der Elektronenstrahl vor der Streukammer in einer Schikane ausgelenkt. Dies bewirkt, dass der Elektronenstrahl schräg in die Streukammer gelangt. In der Streukammer befindet sich ein Separationsmagnet, der den einfallenden Strahl auf das Target lenkt. Die um 180° gestreuten Elektronen werden vom Separationsmagnet, aufgrund der umgekehrten Bewegungsrichtung, zum Spektrometer hin abgelenkt [10]. Zu sehen ist der Aufbau für 180° -Streuung in Abbildung 4.

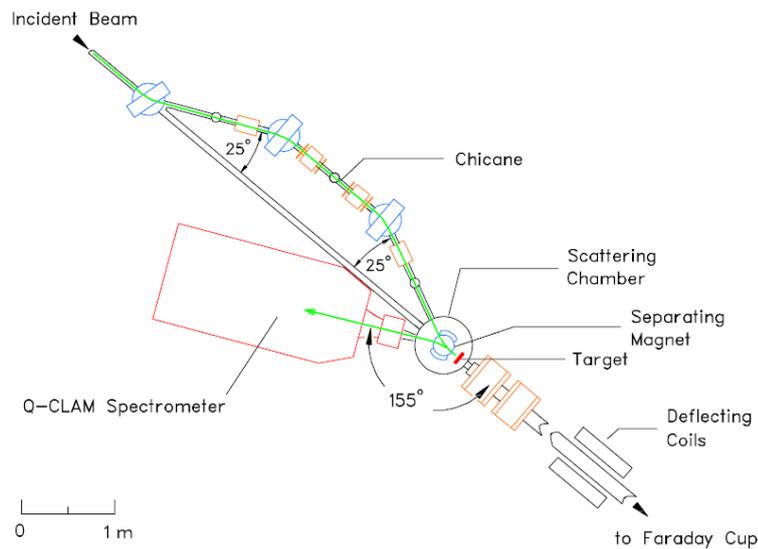


Abbildung 4: Der Aufbau für die Durchführung von 180°-Streuung ist schematisch dargestellt. Die grüne Linie stellt die Flugbahn eines um 180° gestreuten Elektrons dar. [4]

1.4.2. Detektorsystem

Das Detektorsystem besteht aus drei unterschiedlichen Arten von Detektoren. Dabei handelt es sich um Vieldrahtdriftkammern, einen Szintillator und einen Čerenkovzähler. Die Vieldrahtdriftkammern bestehen aus zwei Kathodenfolien, zwischen denen sich eine Ebene mit Anodendrahten befindet. Gefüllt sind die Vieldrahtdriftkammern mit einem Argon-Isobutan Gemisch, welches als Zählgas dient. Zwischen Kathodenfolien und Anodendrahten liegt eine Hochspannung von ca. 6000 V an [4]. Alle Feldlinien die in einem Anodendraht enden, bilden eine Driftzelle.

Die Funktionsweise einer Driftzelle ähnelt der eines Geiger-Müller-Zählrohrs. Durchquert ein gestreutes Elektron die Driftzellen, so entstehen entlang der Elektronentrajektorie Elektron-Ion-Paare. Diese sekundären Elektronen werden entlang der Feldlinien beschleunigt und erzeugen weitere Elektron-Ion-Paare. Als Driftzeit wird die Zeit bezeichnet, die sekundäre Elektronen benötigen, um zu einem Anodendraht zu gelangen. Durch Messen dieser Driftzeiten in benachbarten Driftkammern ist es möglich den Durchstoßpunkt der gestreuten Elektronen durch die Drahtebene zu bestimmen. Die Vieldrahtdriftkammern sind mit 44° gegenüber dem Sollstrahl so geneigt, dass gestreute Elektronen trotz unterschiedlicher Impulse und Eintrittswinkeln in das Spektrometer mehrere Driftzellen durchqueren. Der Sollstrahl entspricht der optischen Achse eines optischen Systems. Durch die Verkippung um 44° wird sichergestellt, dass die Durchstoßpunkte mit ausreichender Genauigkeit bestimmt werden können. In Abbildung 5 sind der schematische Verlauf der Feldlinien und die Flugbahn eines Elektrons durch mehrere Driftzellen abgebildet.

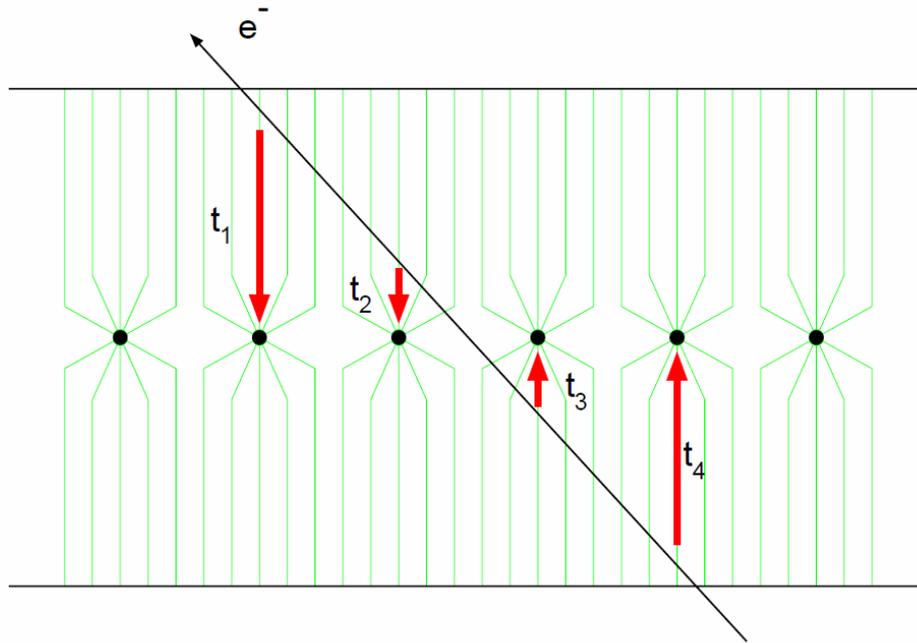


Abbildung 5: Durch Messung der Driftzeiten der sekundären Elektronen kann der Durchstoßpunkt des primären Elektrons durch die Drahtebene bestimmt werden. [4]

Zwei der drei Vieldrahtdriftkammern dienen zur Bestimmung der dispersiven Koordinate X . Daher werden diese Vieldrahtdriftkammern als X_1 und X_2 Vieldrahtdriftkammern bezeichnet. Zur Bestimmung der nicht dispersiven Koordinate Y wird die dritte Vieldrahtdriftkammer verwendet. Ihre Anodendrähte sind gegenüber denen der X_1 und X_2 Vieldrahtdriftkammern um $26,5^\circ$ geneigt. Daher wird diese Vieldrahtdriftkammer nicht als Y sondern als U Vieldrahtdriftkammer bezeichnet. Zusammen mit den Durchstoßpunkten durch die beiden anderen Vieldrahtdriftkammern kann die Koordinate U in eine Y Koordinate umgerechnet werden. Eine direkte Messung der nicht dispersiven Koordinate ist wegen zu großen Spannängen der Anodendrähte nicht möglich. Die Flugbahn eines Elektrons durch alle drei Vieldrahtdriftkammern ist in Abbildung 6 zu sehen.

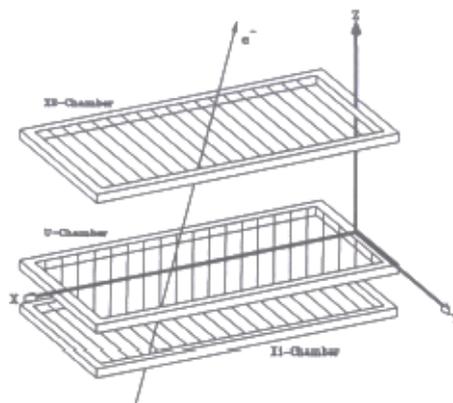


Abbildung 6: Zu sehen sind die drei Drahtebenen sowie die Trajektorie eines Elektrons. [11]

Durch die Bestimmung der drei Durchstoßpunkte ist es möglich die Flugbahn der Elektronen zu rekonstruieren. Dadurch war es bei der Konstruktion des Spektrometers nicht notwendig Abbildungsfehler vollständig zu korrigieren, da dies mit der Elektronentrajektorie

TEIL I:

Entwicklung einer neuen Verfahrsteuerung für das QCLAM-Spektrometer

2. Entwicklung einer neuen Verfahrsteuerung

Bisher erfolgte die Einstellung des Spektrometers für einen vorgegebenen Streuwinkel mit einer Steuerung, die die Einstellung der Fahrtrichtung und der Geschwindigkeit erlaubt und einer Ausgleichsgeraden. Steuerung und Ausgleichsgerade sind in Abbildung 8 bzw. Abbildung 9 zu sehen. Mit einer Betätigung des Schalters „Steuerung“ begann das Spektrometer mit der eingestellten Geschwindigkeit in die angegebene Richtung zu fahren. Ein Druck auf den Schalter „Bremse“ stoppte die Bewegung des Spektrometers.

Diese Art der Bedienung hat zwei Nachteile. Bei der Einstellung eines Winkels muss jedes Mal der Winkel mit Hilfe der Ausgleichsgeraden in einen Kreisbogen umgerechnet werden, welcher an einem Zentimetermaß an der Laufschiene abgelesen werden kann. Zudem könnte beispielsweise ein plötzlich eintretender Notfall während der Bedienung dazu führen, dass der Bedienende sich vom Spektrometer entfernt ohne es angehalten zu haben. Sollte dies eintreten, so würde das Spektrometer in die Strahlführung fahren und schwere Schäden verursachen. Diese Nachteile sollten durch Entwicklung einer neuen Steuerung beseitigt werden.

Die neue Steuerung soll auch die Möglichkeiten bieten nach Eingabe eines Streuwinkels das Spektrometer automatisch zu positionieren. Zusätzlich soll es möglich sein das Spektrometer weiterhin manuell steuern zu können, wobei eine Bewegung nur solange stattfinden soll wie der Knopf gedrückt wird. Endschalter sollen dafür sorgen, dass das Spektrometer nicht zu weit fahren kann, und somit die Betriebssicherheit weiter erhöhen.



Abbildung 8: Zu sehen ist die alte Steuerung. Sie verfügt über zwei Knöpfe zum Fahren und Anhalten. Die Geschwindigkeit kann mit einem Drehregler eingestellt werden. Ein Zwei-Positionsschalter dient zum Auswählen der Fahrtrichtung.

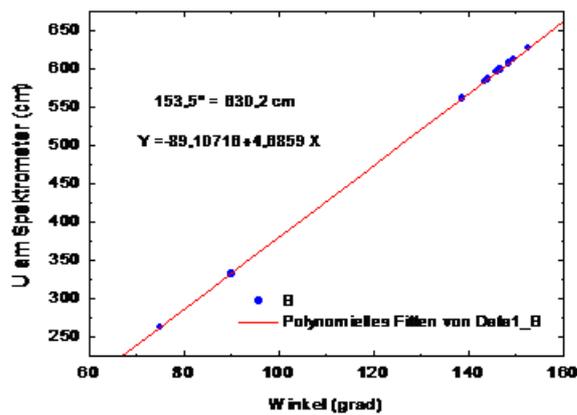


Abbildung 9: Abgebildet ist die Ausgleichsgerade, mit welcher die Streuwinkel in einen Kreisbogen umgerechnet werden. [13]

2.1. Verwendete Komponenten

Am Spektrometer war bereits ein Drehgeber angebracht, welcher ursprünglich für die Positionierung des Spektrometers gedacht war. Dieser Drehgeber musste jedoch wegen eines nicht reparierbaren Defekts ersetzt werden. Der neue Drehgeber stammt von der Firma Kübler und trägt die Bezeichnung 8.3620.543E.0360. Dabei handelt es sich um einen optischen Inkrementalgeber mit einer Hohlwelle und einer Auflösung von 1000 Schritten pro Umdrehung. Eine Abschätzung der Größen der Zahnräder, des Rades und des Radius der Laufschiene hat eine Streuwinkelauflösung von ca. $0,004^\circ$ ergeben. Es ist jedoch davon auszugehen, dass die Kalibrierung diese Genauigkeit nicht erreicht. Zudem ist eine Positionierung mit einer solchen Präzision aufgrund des großen Öffnungswinkels des Spektrometers nicht erforderlich. Der Drehgeber wird mit einem Halterungsadapter in der alten Drehgeberhalterung, welche für die Aufnahme des neuen Drehgebers angepasst werden musste, befestigt. Dies ist in Abbildung 10 zu sehen.

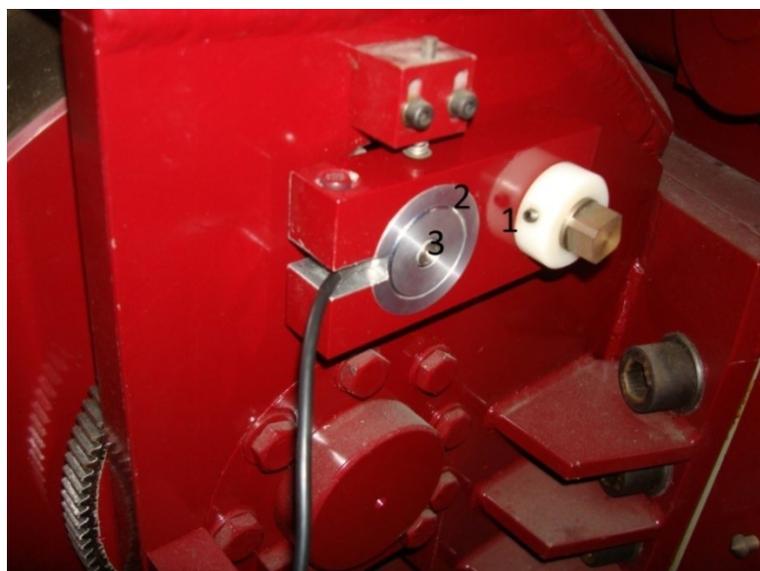


Abbildung 10: Zu sehen sind die alte Halterung (1), der Halterungsadapter (2) und der neue Drehgeber (3). Der Drehgeber ist über ein Zahnrad mit dem großen Zahnrad und somit mit einem Rad des QCLAM-Spektrometers verbunden.

Um möglichst wenig am System ändern zu müssen, wurde das Zahnrad, welches am alten Drehgeber befestigt war, weiter verwendet. Dafür wurde eine passende Welle in der institutseigenen Mechanikwerkstatt angefertigt.

Der Bereich, in dem das Spektrometer gefahren werden kann ohne Schäden am Strahlrohr zu verursachen, wird durch zwei Endschalter eingegrenzt. Verwendet werden die Endschalter von Pizzato Electrica mit der Bezeichnung FR 531-M2. Diese werden ab einer Kraft von 0,25 N ausgelöst. Befestigt werden die Endschalter an der Laufschiene des Spektrometers mit Klemmen, siehe Abbildung 11, die in der institutseigenen Mechanikwerkstatt angefertigt wurden.



Abbildung 11: Die Endschalter werden mit Klemmen, die in der institutseigenen Mechanik Werkstatt konstruiert wurden, an der Laufschiene befestigt.

Um die Signale des Benutzers sowie des Drehgebers und der Endschalter verarbeiten zu können, wird ein Multifunktionsdisplay (MFD) vom Typ MFD TITAN von Eaton verwendet. Das Multifunktionsdisplay besteht aus einem Display vom Typ MFD-80, einem CPU-Modul des Typs MFD-CP10-NT und einem Modul für Ein- und Ausgänge des Typs MFD-TA17. Das Modul für Ein- und Ausgänge verfügt über zwölf Eingänge und vier Ausgänge. Jeweils vier Eingänge sind für hohe Frequenzen digitaler Signale (Eingänge 1, 2, 3, 4) beziehungsweise analoge Signale (Eingänge 7, 8, 11, 12) geeignet. Analoge Signale liegen zwischen 0 V und 10 V und können mit einer Auflösung von 10 Bit verarbeitet werden. Um die Eingabe des gewünschten Winkels möglichst einfach zu gestalten, fiel die Wahl auf ein Wendepotentiometer, welches mit einem Analogeingang verbunden ist [14]. Eine Auflistung aller verwendeten Teile ist im Anhang ab Seite 42 zu finden.

2.2. Speicherprogrammierbare Steuerung

Vor Einführung der speicherprogrammierbaren Steuerung (SPS) erfolgte die Steuerung von Maschinen mit festverdrahteten Relaisschaltungen [15]. Die festverdrahteten Relaisschaltungen sind kompliziert und Änderungen an der Arbeitsweise sind aufwändig. Bei einer SPS handelt es sich um einen simplen Computer, der eingehende Signale verarbeitet und nach Verarbeitung dieser Signale Steuersignale ausgibt [16]. Programmiert wird die Steuerung auf handelsüblichen Computern mit der entsprechenden Entwicklungsumgebung. Nach

Fertigstellung der Programmierung kann das Steuerungsprogramm per USB-Kabel auf die SPS übertragen werden.

2.3. Steuerkasten

Die elektrischen Komponenten für die Steuerung wurden in einen Steuerkasten integriert. Im Steuerkasten sind die SPS sowie alle Bedienelemente untergebracht, das Netzteil befindet sich außerhalb des Steuerkastens. Der Steuerkasten, welcher in Abbildung 12 zu sehen ist, besteht aus zwei zusammengeschraubten Kästen. Im großen Kasten sind das Multifunktionsdisplay, ein Drehregler für die Winkelvorgabe und ein Dreipositionsschalter, mit welchem es möglich ist, die Fahrtrichtung des manuellen Modus einzustellen bzw. der automatische Modus aktiviert werden kann, untergebracht. Der kleine Kasten beherbergt den Not-Aus-Taster und drei weitere Knöpfe. Die Funktionen der drei Knöpfe sind von oben nach unten Reset, Start/Stopp und Geschwindigkeitsauswahl. An den Kabelaustrittsöffnungen des großen Kastens sind Zugentlaster angebracht.

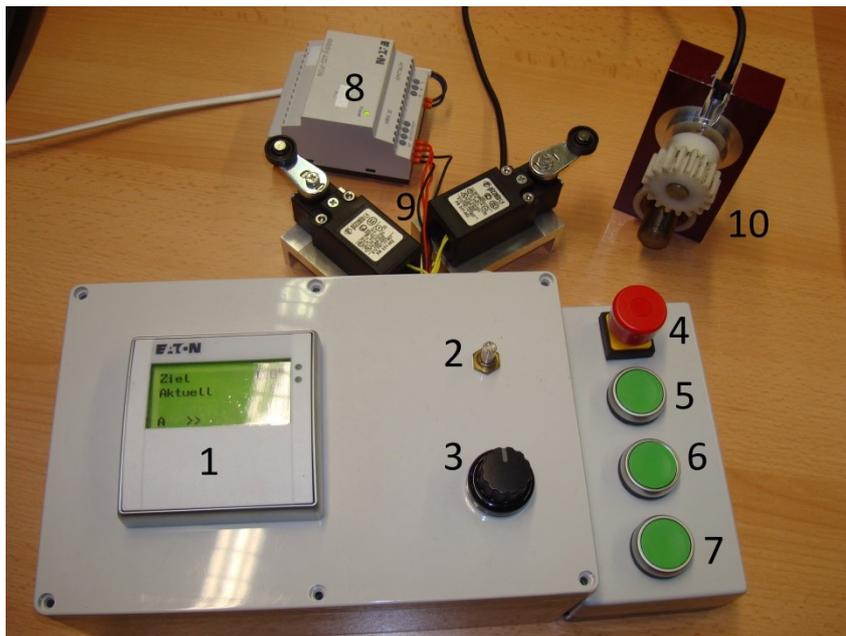


Abbildung 12: Abbildung des neuen Steuerkastens. (1) Multifunktionsdisplay, (2) Wendelpotentiometer, (3) Dreipositionsschalter, (4) Not-Aus-Taster, (5) Start/Stopp, (6) Geschwindigkeitsauswahl, (7) Reset, (8) Netzteil, (9) Endschalter, (10) Drehgeber mit Halterung.

2.4. Funktionsweise des Steuerungsprogramms und Bedienung

Das Steuerprogramm lässt sich in zwei unterschiedliche Betriebsarten, den automatischen und den manuellen Modus, aufteilen. Zum Umschalten zwischen den Betriebsarten dient der Dreipositionsschalter. Befindet sich der Schalter auf der mittleren Position, so ist der automatische Modus aktiv. Der manuelle Modus wird aktiviert sobald explizit eine Fahrtrichtung ausgewählt wird.

Im automatischen Modus dient zur Einstellung des gewünschten Winkels ein Potentiometer, welches mit einem Analogeingang der SPS verbunden ist. Im Gegensatz zum Potentiometer liefert der Drehgeber kein analoges Signal, sondern Rechtecksignale. Diese werden von einem Hochgeschwindigkeitszähler gezählt und in einen Winkel umgerechnet. Anschließend wird überprüft, ob der eingestellte Winkel kleiner oder größer als der aktuelle Winkel ist. Dadurch wird die Bewegungsrichtung des Spektrometers bestimmt und eingestellt.

Ein Druck auf den Start-/Stopp-Knopf setzt das Spektrometer in Bewegung. Manuell kann das Spektrometer durch erneutes Drücken des Start-/Stopp-Knopfes angehalten werden. Das Spektrometer stoppt automatisch, wenn der Endschalter, welcher sich in Bewegungsrichtung befindet, betätigt wird oder der Zielwinkel erreicht wird.

Der manuelle Modus ermöglicht das Fahren des Spektrometers, ohne zuvor ein Ziel eingeben zu müssen. Die Bewegungsrichtung kann mit einem Dreipositionsschalter eingestellt werden. Das Spektrometer fährt in diesem Modus so lange man den Start-/Stopp-Knopf gedrückt hält, oder der entsprechende Endschalter betätigt wird.

Ein Reset des Hochgeschwindigkeitszählers erfolgt durch den Reset-Knopf, dabei wird die aktuelle Position auf den Winkel gesetzt, der mit dem Potentiometer vorgegeben wurde. Mit dem Geschwindigkeitsauswahl-Knopf ist es möglich, zwischen zwei Geschwindigkeiten zu wählen. Standardmäßig ist die langsame Geschwindigkeitsstufe eingestellt. Die Geschwindigkeit kann nur während des Fahrens verändert werden, ein Stopp des Spektrometers setzt die eingestellte Geschwindigkeit auf langsam zurück.

Nachdem das Spektrometer gefahren wurde bleibt der Hochgeschwindigkeitszähler noch drei Sekunden aktiv. In dieser Zeit kann das Spektrometer vollständig zum Stehen kommen. Während das Spektrometer nicht gefahren wird, können auch am Drehgeber Wartungsarbeiten durchgeführt werden, ohne den aktuellen Winkel zu verfälschen. Beim Anfahren im automatischen Modus gibt es eine Verzögerung von 0,5 Sekunden. Diese Verzögerung wurde eingebaut, da Schwankungen am Analogeingang mehrere Richtungswechsel pro Sekunde verursachen können. Die Fahrtrichtung wird im automatischen Modus direkt nach Betätigung des Start-/Stopp-Knopfes festgelegt. Die Verzögerung sorgt dafür, dass die Festlegung der Fahrtrichtung und das Losfahren nicht gleichzeitig erfolgen. Des Weiteren stellt das Steuerprogramm sicher, dass nur der Endschalter in Bewegungsrichtung des Spektrometers eine Weiterfahrt in diese Richtung verhindert. Somit kann das Spektrometer nicht über den Endschalter hinausfahren, aber vom ausgelösten Endschalter weggefahren werden. Um die Betriebssicherheit weiter zu erhöhen sind die Endschalter und der Not-Aus-Taster so verkabelt, dass die Stromkreise im Normalbetrieb geschlossen sind. Sollte sich eines dieser Kabel lösen wird dies als eine Betätigung des entsprechenden Schalters registriert.

2.5. Anzeigen auf dem Display

Das Display des Steuerkastens verfügt über zwei Masken, eine für den automatischen Modus und eine für den manuellen Modus. Diese beiden Masken unterscheiden sich, wie in Abbildung 13 zu sehen ist, nur geringfügig voneinander. Der Buchstabe „A“ bzw. „M“ in der Ecke links unten weist auf den automatischen bzw. auf den manuellen Modus hin.

Im automatischen Modus wird der mit dem Potentiometer eingestellte Winkel als „Ziel“ bezeichnet, „Vorgabe“ ist die Bezeichnung dieses Winkels im manuellen Modus. Unterhalb dieses Winkels weist die Beschriftung „Aktuell“ auf die aktuelle Position des Spektrometers hin. Zur Anzeige der Geschwindigkeit dienen die zwei bzw. vier Pfeile auf dem Display. Zwei Pfeile stehen für die langsame und vier Pfeile für die schnelle Geschwindigkeitsstufe. In der Ecke rechts unten erscheint bei Betätigung der Endschalter ein blinkender Warnhinweis, welcher sich aus „End“ für Endschalter und „R“ bzw. „L“ für den rechten bzw. linken Endschalter zusammensetzt. Gleichzeitig beginnt neben dem Display die rote Leuchtdiode zu leuchten. Die rote Leuchtdiode beginnt auch zu leuchten, wenn der Not-Aus-Taster betätigt wird. Zusätzlich wird das Display abgedunkelt und ein blinkender Schriftzug weist auf den betätigten Not-Aus-Taster hin.

Im manuellen Modus wird zudem „L/R“ für die Fahrtrichtung angezeigt, dabei blinkt der Buchstabe für die eingestellte Richtung.



Abbildung 13: Abgebildet sind die Masken des automatischen (links) und des manuellen (rechts) Modus. Die Masken wurden im Simulationsmodus der Entwicklungssoftware aufgenommen. Bei der linken Maske sollte das Spektrometer mit der schnellen Geschwindigkeitsstufe zu einem Winkel von 100° fahren, daher leuchtet die Grüne LED. Die Abbildung der rechten Maske entstand, während im manuellen Modus die Fahrtrichtung links ausgewählt war. Zusätzlich wurden der linke Endschalter und der Not-Aus-Taster betätigt, deshalb leuchtet auch die rote LED und das Display wurde abgedunkelt.

TEIL II:

Überarbeitung und Erweiterung der Stromauslese am QCLAM-Spektrometer

3. Strom- und Ladungsauslese am Faraday-Cup

Für die Strahloptimierung sind an mehreren Stellen Faraday-Cups in der Strahlführung, mit deren Hilfe der Strahlstrom gemessen werden kann, angebracht. Auch hinter den Spektrometern Lintott und QCLAM befinden sich Faraday-Cups, welche als Strahlfänger dienen und ebenfalls für die Strahloptimierung benutzt werden.

Diese beiden Faraday-Cups sind auch für die Auswertung der Messdaten wichtig, da die während des Experiments gesammelte Ladung zur Berechnung der experimentellen differentiellen Wirkungsquerschnitte (Gleichung (1.6)) benötigt wird.

Somit muss der Strom an den Faraday-Cups sowohl im Kontrollraum für die Strahloptimierung als auch im Messraum für die Datenerfassung des Experiments bereitstehen. Um dies zu realisieren wird die QM07-Elektronik verwendet, die im Rahmen einer Dissertation am Institut für Kernphysik der TU Darmstadt entwickelt wurde [17]. Diese Elektronik diente bereits zur Erfassung des Stroms am Faraday-Cup hinter dem Lintott-Spektrometer [18].

3.1. QM07-Elektronik und Aufbau

Die QM07-Elektronik besteht aus einem Crate mit der QM07-Hauptplatine, einer Kontrolleinheit und einer QM07-Prozessor-Einheit, zusätzlich können bis zu 13 Einschübe eingebaut werden. Die Kontrolleinheit besitzt zwei CAN-Anschlüsse (Controller Area Network), welche das Anschließen an einen Computer ermöglichen. Für die Erfassung der Ströme an den Faraday-Cups wird ein IADC-Einschub mit zwei Messkanälen verwendet. Das I in IADC steht für Strom, ADC ist die Abkürzung für „analog to digital converter“. Folglich digitalisiert der IADC-Einschub den gemessenen Strom. Dieser Einschub kann Ströme zwischen $-101,6 \mu\text{A}$ und $101,6 \mu\text{A}$ messen. Eine Kalibrierungseinheit stellt Ströme von $1 \mu\text{A}$ und $100 \mu\text{A}$, sowie eine Spannung von 10 V mit wahlweise positiver oder negativer Polarität bereit. Diese Ströme können für Testmessungen an den IADC-Einschub angeschlossen werden [19].

In Abbildung 14 ist das verwendete Crate zu sehen. Der obere CAN-Anschluss der Kontrolleinheit ist mit einem handelsüblichen Computer verbunden. Dieser Computer dient auch für die Datenaufnahme des Lintott-Spektrometers. Die beiden Messkanäle des IADC-Einschubs sind über Koaxialkabel mit den Faraday-Cups hinter dem Lintott-Spektrometer (oberer Messkanal) und dem QCLAM-Spektrometer (unterer Messkanal) verbunden. Die bisherige QM07-Prozessor-Einheit wurde durch eine mit einem zusätzlichen Aufsatz, der über drei NIM-Ausgängen verfügt, ansonsten baugleich ist, ausgetauscht. Am mittleren Ausgang des Aufsatzes wird der frequenzgewandelte Strom ausgegeben. Dieser Ausgang ist mit einem Scaler verbunden, der in die Datenaufnahme des QCLAM-Spektrometers eingebunden ist. Die maximale Frequenz des frequenzgewandelten Stroms liegt bei $32,768 \text{ MHz}$ und wird bei einer Stromstärke von $101,6 \mu\text{A}$ erreicht, das Vorzeichen des Stroms ist dabei nicht von Bedeutung.

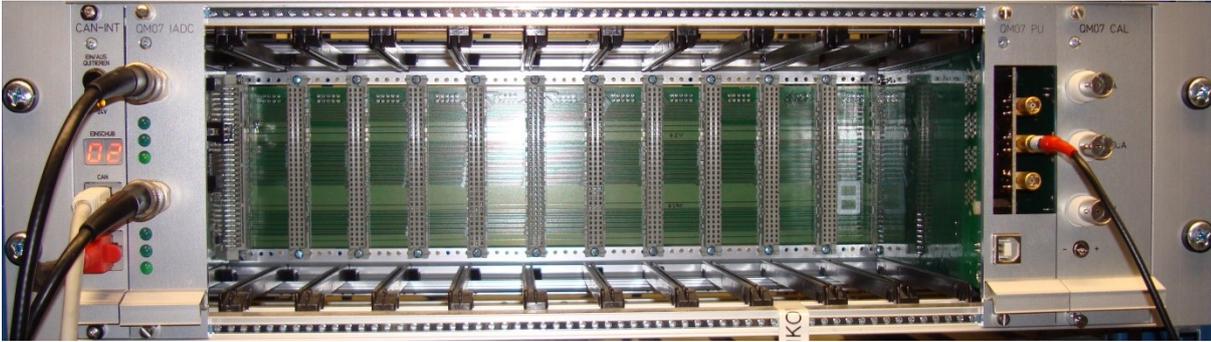


Abbildung 14: Zu sehen ist das QM07 Crate mit den verwendeten Einschüben. Im Crate befinden sich von links nach rechts die Kontrolleinheit, der IADC-Einschub, die neue Prozessor-Einheit und die Kalibrierungseinheit.

3.2. Datenauslese der QM07-Elektronik mit EPICS

EPICS (Experimental Physics and Industrial Control System) ermöglicht es Experimente mit input/output Controllern (IOCs) fernzusteuern und Daten über ein Netzwerk zur Verfügung zu stellen [20]. Mit IOCs kann die Kommunikation zwischen Computern und Umsetzern, die wiederum Geräte ansteuern können, erfolgen. Die grafischen Benutzeroberflächen auf den Computern werden operator interfaces (OPIs) genannt, sie werden mit Control System Studio (CSS) [21] erstellt. Weitere Informationen sind in [22] zu finden.

Da sowohl ein IOC als auch ein OPI für die Stromauslese des Faraday-Cups hinter dem Lintott-Spektrometer bereits existierten [23], wurden diese überarbeitet und erweitert. Der IOC verwendet nun ein neueres Template für die Stromauslese und liest die Ströme an beiden Messkanälen aus. Das alte Template wird nicht mehr weiterentwickelt und wurde nur für die Kompatibilität zu älteren IOCs nicht gelöscht. Erweitert wurde der IOC um die Auslese der Ladungen der Faraday-Cups und den über eine Sekunde gemittelten Strom. Dieser gemittelte Strom kann in der Beschleunigerdatenbank gespeichert werden. Um den benötigten Speicherplatz zu reduzieren überprüft der IOC ob der gemittelte Strom sich um einen vorgegebenen Wert von dem zuletzt gespeicherten gemittelten Strom unterscheidet, dadurch wird das Rauschen unterdrückt.

Die im Folgenden vorgestellten Gleichungen und die Umrechnung der Stromwerte sind [18] entnommen. Der aktuelle Strom kann mit der Gleichung (3.1) bzw. (3.2) berechnet werden. Für positive Ströme muss Gleichung (3.1) verwendet werden, negative Ströme werden hingegen mit Gleichung (3.2) berechnet.

$$I_+ = I_{\text{dez}} \cdot \frac{101,6 \cdot 10^{-6} \text{ A}}{2^{28} - 1} \quad (3.1)$$

$$I_- = (2^{32} - I_{\text{dez}}) \cdot \frac{-101,6 \cdot 10^{-6} \text{ A}}{2^{28}} \quad (3.2)$$

I_{dez} ist die dezimale Darstellung, der von der QM07 ausgegebenen 32-Bit Hexadezimalzahl. 29-Bit entsprechen dem Vollausschlag, der bei einer Stromstärke von 101,6 μA erreicht wird.

Die Berechnung der Ladung erfolgt mit den Gleichungen (3.3) bis (3.5). I_{Mittel} bezeichnet hierbei den über interne Messzyklen der QM07-Elektronik gemittelten Strom und nicht den von EPICS gemittelten Strom, der in der Beschleunigerdatenbank gespeichert werden kann.

$$I = I_{\text{dez}} \cdot 256 \cdot \frac{101,6 \cdot 10^{-6} \text{ A}}{2^{28}} \quad (3.3)$$

$$I_{\text{mittel}} = \frac{I_{\text{Ende}} - I_{\text{Start}}}{N_{\text{Ende}} - N_{\text{Start}}} \quad (3.4)$$

$$Q = I_{\text{mittel}} \cdot (T_{\text{Ende}} - T_{\text{Start}}) \quad (3.5)$$

Für die Berechnung des Stroms I ist die Multiplikation mit 256 notwendig, da hier I_{dez} aus einer 56-Bit Hexadezimalzahl berechnet wird. Nach der Multiplikation ist I_{dez} auf 64-Bit aufgewertet, wobei positive und negative Zahlen jeweils 32-Bit besetzen. Die Gesamtmessdauer wird als T bezeichnet, N steht für die Anzahl der internen Messzyklen.

Für die Anzeige des Stroms an den Faraday Cups stehen drei neue OPIs zur Verfügung. Dabei handelt es sich um zwei kleinere OPIs (`lintott_cup_widget.opi` und `qclam_cup_widget.opi`) für die beiden Faraday-Cups, sowie um ein OPI (`spektrometer_cup.opi`), welches umfangreiche Informationen zu beiden Faraday-Cups und dem QM07 Crate bietet.

Zu sehen sind die beiden kleineren OPIs in Abbildung 15. Die beiden kleinen OPIs zeigen den aktuellen Strom an den Faraday-Cups an und bieten die Möglichkeit, einen Grenzwert für den Strahlstrom einzustellen (das Vorzeichen muss bei der Eingabe berücksichtigt werden). Dies ermöglicht dem Experimentator eine möglichst deutliche optische Warnung, falls der Strahlstrom den Grenzwert überschreitet. Wenn dies eintritt, färbt sich der Hintergrund rot. Der Reset Knopf setzt den Grenzwert auf 1000 nA. Als Vorlage für das OPI `spektrometer_cup.opi` dienten das OPI `Lintott.opi` [23] und das OPI `LintottCup.opi` [24]. In den Tabs `Lintott` und `QClam` sind `lintott_cup_widget.opi` bzw. `qclam_cup_widget.opi` eingebunden. Zusätzlich werden in jeweils zwei Graphen der gemessene Strom über einen Zeitraum von ca. einer Minute bzw. einer Stunde angezeigt. Dies ist für die Überwachung von Experimenten hilfreich, da ein einzelnes Spektrum üblicherweise über eine Stunde aufgenommen wird. In dem Tab „Überblick“ werden für beide Faraday-Cups der gemessene Strahlstrom, der von EPICS gemittelte Strahlstrom und die gesammelte Ladung angezeigt. Abgebildet sind die drei Ansichten von `spektrometer_cup.opi` in den Abbildungen Abbildung 16 bis Abbildung 18.

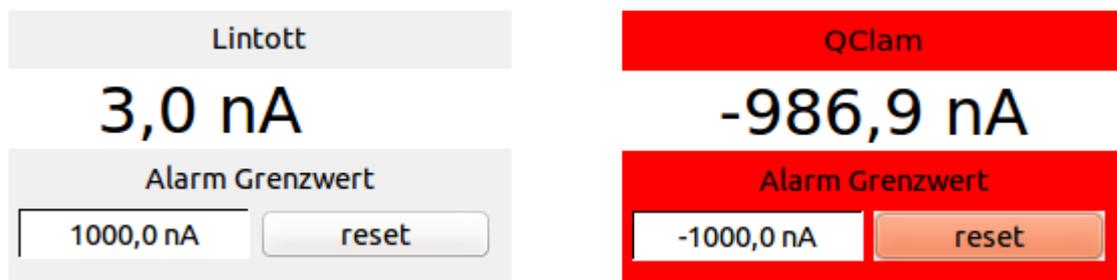


Abbildung 15: Links ist das operator interface für das Lintott-Spektrometer zu sehen. Auf der rechten Seite befindet sich das operator interface für das QCLAM-Spektrometer, hier ist der Hintergrund rot gefärbt, da der Strom den Grenzwert überschritten hat. Anstelle des Strahlstroms am Faraday-Cup hinter dem QCLAM-Spektrometer wurde die Kalibrierungseinheit der QM07-Elektronik verwendet.

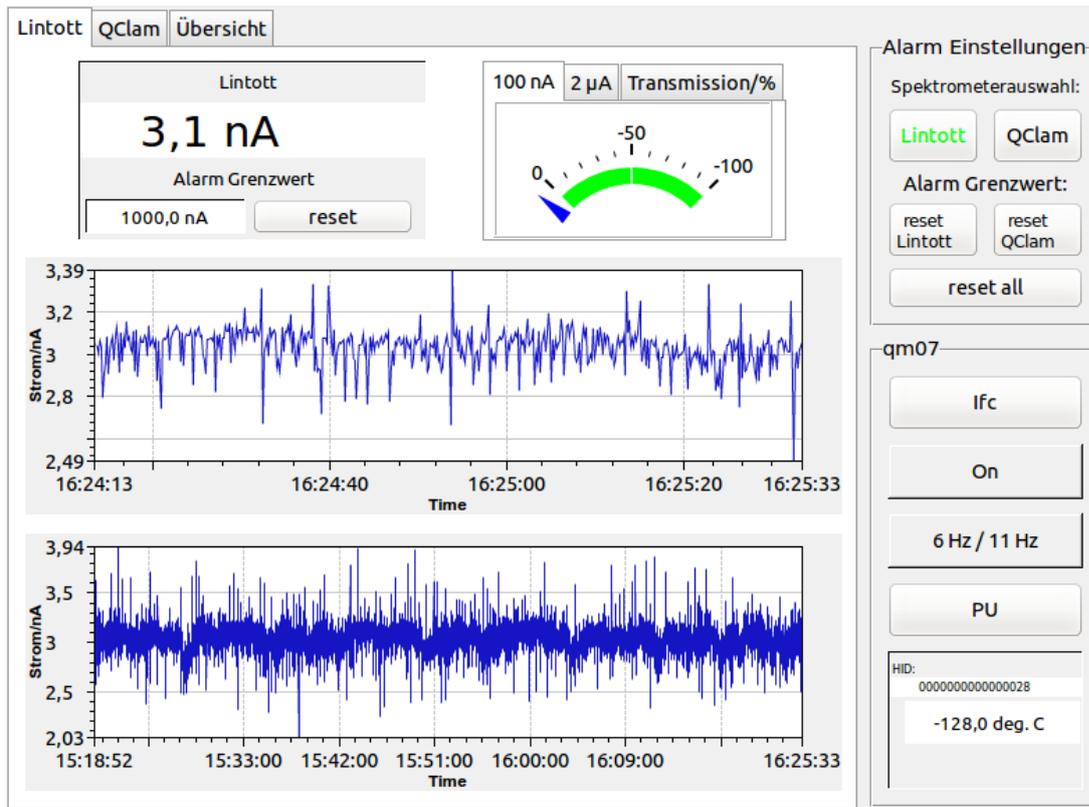


Abbildung 16: Im Abschnitt „Lintott“ werden Informationen zum Strahlstrom am Lintott-Spektrometer angezeigt. Auch die Vorgabe eines Grenzwertes für den Strahlstrom kann hier erfolgen. Für die Überwachung von Experimenten sind die beiden Graphen hilfreich.

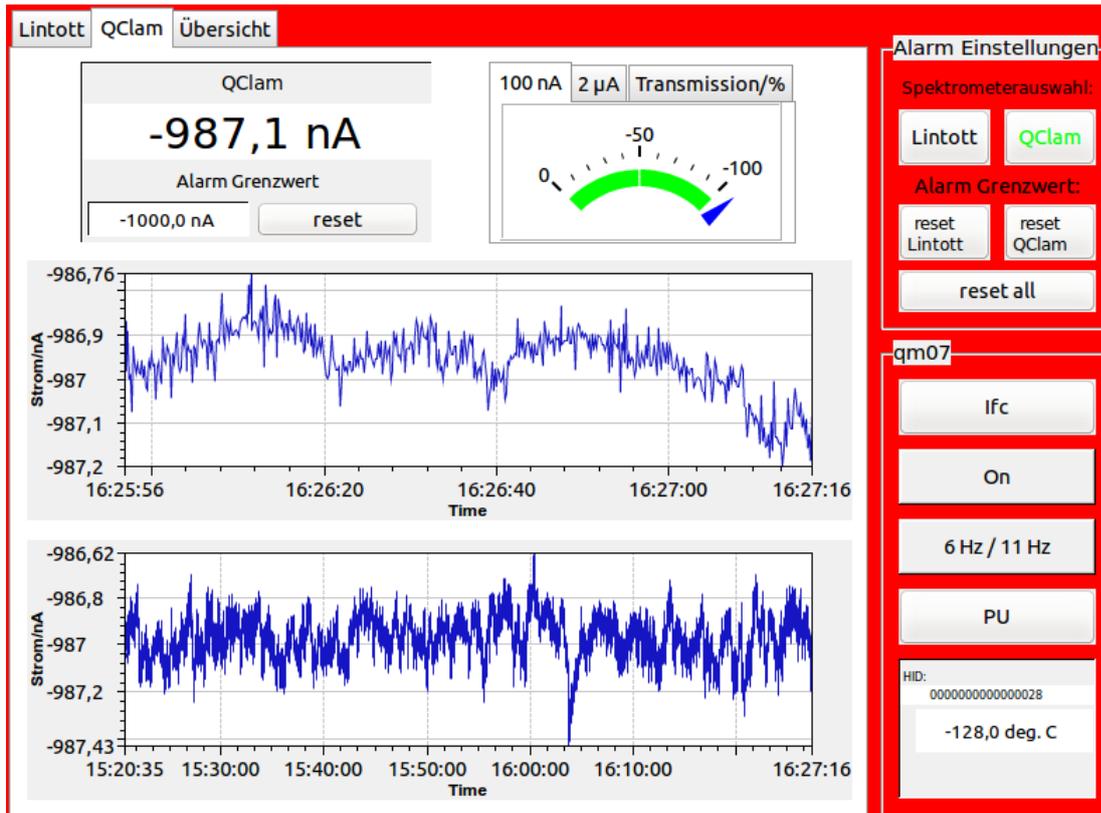


Abbildung 17: Der Abschnitt „QClam“ ist aufgebaut wie der Abschnitt „Lintott“. Der Hintergrund ist rot gefärbt, da der eingestellte Grenzwert überschritten wurde.

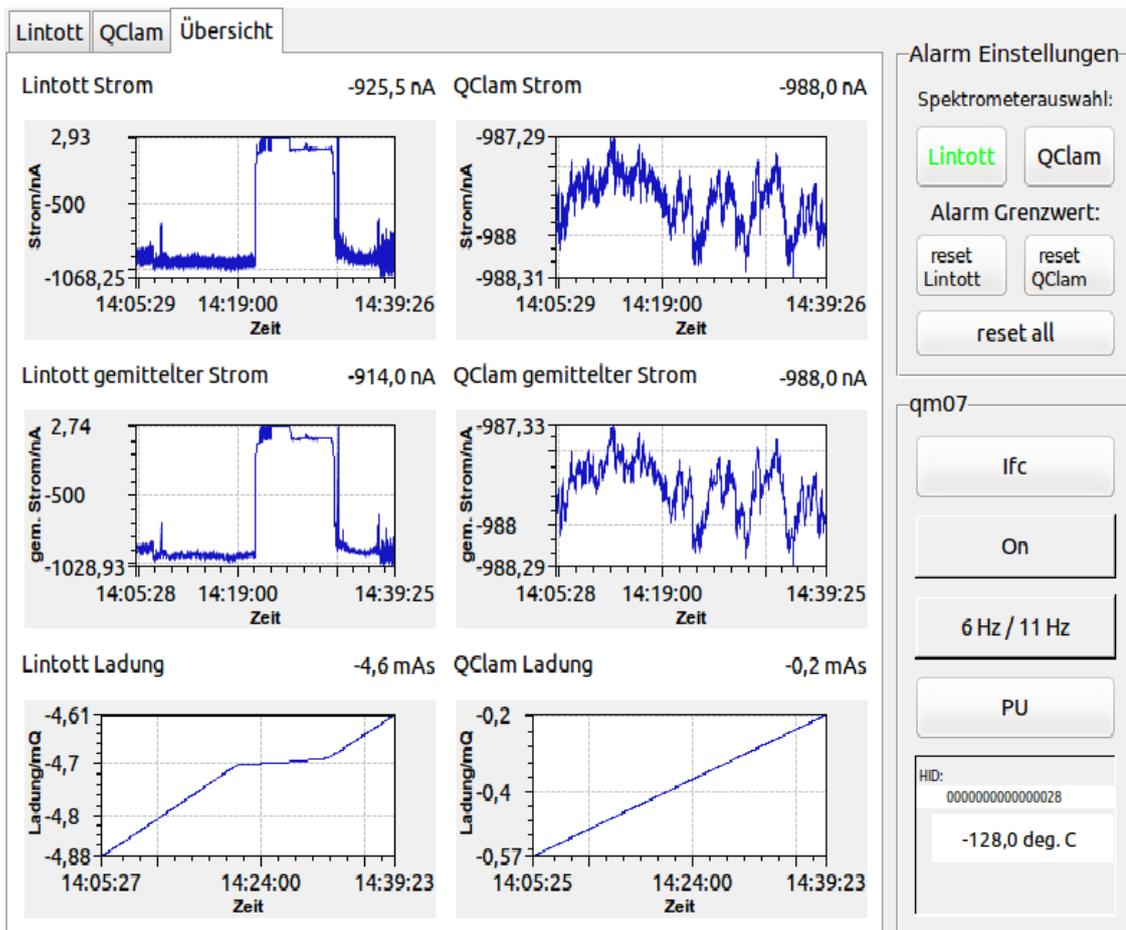


Abbildung 18: Im Abschnitt „Übersicht“ sind der gemessene Strom, der gemittelte Strom und die gemessene Ladung für beide Faraday-Cups zu sehen. Während der Aufnahme dieses Bildes wurden Strahloptimierungen für Experimente am Lintott-Spektrometer durchgeführt, statt dem Faraday-Cup hinter dem QCLAM-Spektrometer wurde der $1 \mu\text{A}$ Testausgang der Kalibrierungseinheit der QM07-Elektronik verwendet.

3.3. Datenaufnahme und Aufbau

Die Datenaufnahme des QCLAM-Spektrometers erfolgt mit dem Multi Branch System (MBS), das von der Gesellschaft für Schwerionenforschung (GSI) entwickelt wurde. Gespeichert werden die Daten in binär kodierten Listmode Dateien. Diese können mit dem, ebenfalls von der GSI entwickelten, GSI Object Oriented On-line Off-line system (Go4) analysiert werden [25]. Bei jedem Auslesevorgang wird unter anderem ein Zeitstempel gespeichert. Er setzt sich aus der Unixzeit und Millisekunden zusammen. Weitere Informationen zu dem Umgang mit MBS sind in [26] zu finden. Eine Installationsanleitung für Go4 unter Linux und Windows wurde im Rahmen diese Arbeit angefertigt und befindet sich im Beschleunigerwiki des Instituts für Kernphysik der TU Darmstadt [27]. Die von den Entwicklern bereitgestellte Anleitung befindet sich auf der Go4 Seite der GSI [25].

3.3.1. Aufbau der Datenaufnahme

An der äußerst linken Position im Crate befindet sich das Controller Modul Rio3. Auf diesem Modul läuft MBS. Rechts neben dem Controller Modul befindet sich das Trigger Modul Triva. Ein vom Trigger Modul registriertes Trigger-Signal wird an das Controller Modul weitergeleitet und dort von MBS interpretiert [26]. Für die Erfassung des Strahlstroms wird

der Scaler SIS 3820 von Struck verwendet. Die maximale Zählrate des Scalers liegt bei 250 MHz [28] und ist somit deutlich größer als die Maximalfrequenz des frequenzgewandelten Stromes der QM07-Elektronik. Dieser Scaler verfügt jedoch nur über ECL Eingänge, daher wird das, von der QM07 Elektronik stammende, Signal von einem NIM/ECL Übersetzer konvertiert. Abgebildet ist der Aufbau in den Abbildungen Abbildung 19 und Abbildung 20.

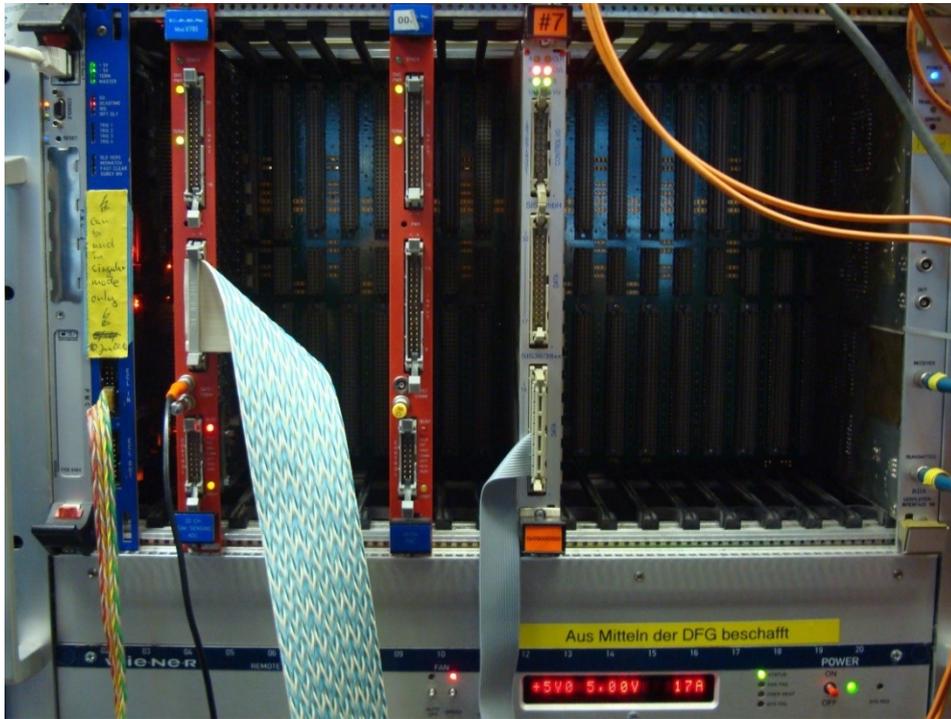


Abbildung 19: Im äußersten linken Steckplatz befindet sich die Rio, rechts daneben ist das Triggermodul Triva untergebracht. Die beiden roten Module sind ein ADC und ein TDC, sie sind jedoch für die Stromauslese ohne Bedeutung. Der SIS 3820 Scaler ist das graue Modul in der Mitte. Der äußerste rechte Steckplatz ist von einem Modul belegt, welches zur Datenaufnahme der Driftkammern dient.

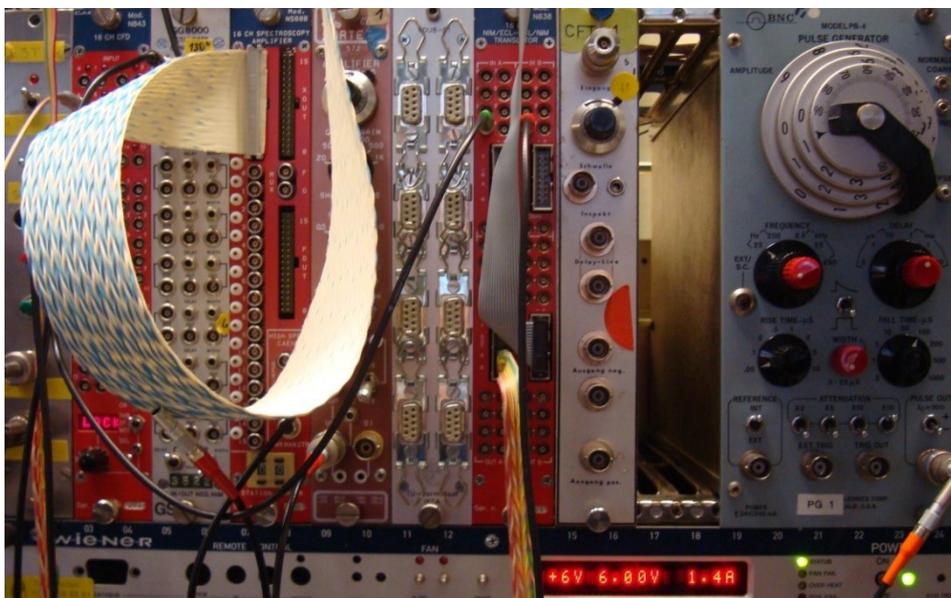


Abbildung 20: Der verwendete NIM/ECL Übersetzer ist das rote Modul in den Steckplätzen 13 und 14. Das Kabel vom QM07 Crate ist im Kanal 0 im Abschnitt „IN B“ angeschlossen. Das graue Kabel leitet das übersetzte Signal an den SIS 3820 Scaler weiter.

3.3.2. Datenaufnahme und Testmessung

Die Verarbeitung und Interpretation von Trigger-Signalen wird in der Datei `f_user.c` festgelegt [26]. Als Vorlage diente ein Projekt, welches unter anderem den SIS Scaler verwendete. Aus diesem Projekt wurden die Dateien `f_user.c` sowie zwei Header für die Adressierung des Scalers verwendet [29]. Die anderen benötigten Dateien wurden aus dem Silizium Ball Projekt entnommen [26, 30].

Da in der `f_user.c` Datei nicht benötigte Module verwendet wurden und der Scaler in einem Preset Modus für eine genauere Zeitmessung, als es MBS ermöglicht, betrieben wurde, mussten einige Anpassungen vorgenommen werden. Im ersten Schritt wurden alle nicht benötigten Module entfernt. Der zweite Schritt bestand darin den Scaler in einen Betriebsmodus zu versetzen, in dem er Rechtecksignale zählt, was im Preset Modus nicht möglich war. Abschließend wurde im dritten Schritt der Code übersichtlicher gestaltet. Als Vorbild diente das Silizium Ball Projekt [26, 30]. Dabei wurde eine zusätzliche Datei `sis_3820.c` erstellt. In dieser Datei wurden Funktionen für das Initialisieren und Auslesen des Scalers definiert. Eine genauere Beschreibung des Vorgehens ist im Anhang ab Seite 51 zu finden.

Eine Testmessung erfolgte mit dem oben beschriebenen Aufbau. Das Trigger Signal lieferte ein Pulser mit einer schwankenden Frequenz von ca. 20 bis 26 Hz. An der QM07-Elektronik wurde der untere Messkanal mit dem 1 μA Ausgang der Kalibrierungseinheit verbunden. Der anliegende Strom schwankte während der ca. 10 Minuten dauernden Messung laut der Stromauslese mit EPICS zwischen $-987,3 \text{ nA}$ und $-987,6 \text{ nA}$.

In Abbildung 21 sind die gemessenen Werte des Scalers über die Anzahl der Auslesevorgänge aufgetragen. Die Werte des Scalers in Abhängigkeit von der Zeit, die von MBS automatisch bei jedem Auslesevorgang gespeichert wird, sind in Abbildung 22 zu sehen. Dies entspricht jedoch nicht dem, für einen konstanten Strom, erwarteten Verlauf einer Geraden. Der Vorzeichenwechsel deutet auf einen Speicherüberlauf hin. Dies ist jedoch in Abbildung 21 nicht zu sehen. Auffällig ist auch, dass einige Balken kleiner als die vorherigen sind, was einer negativen Zählrate entsprechen würde. Eine negative Zählrate ist in Abbildung 21 ebenfalls nicht zu sehen. Weiterhin fällt auf, dass die Anzahl der Balken sehr viel kleiner ist, als die Anzahl der Auslesevorgänge. Auch der Abstand von ca. 10 Sekunden zwischen zwei benachbarten Balken ist auffällig.

Die Ursache für die Abweichung vom erwarteten Verlauf wird deutlich, wenn die Zeit über die Auslesevorgänge aufgetragen wird. Dies ist in Abbildung 23 abgebildet. Wie zu sehen ist, ist der Zeitstempel für ca. 10 Sekunden konstant und macht dann einen Sprung von ca. 10 Sekunden. Dadurch kann die Abweichung von der erwarteten Gerade erklärt werden. Alle Auslesevorgänge des Scalers mit gleichem Zeitstempel werden zu einem einzigen Wert aufaddiert, der dieser Zeit zugeordnet ist. Aufgrund der schwankenden Frequenz des nicht idealen verwendeten Pulsers, können in Intervallen mit gleichem Zeitstempel unterschiedlich viele Auslesevorgänge liegen, wodurch ein Balken kleiner als der vorherige werden kann.

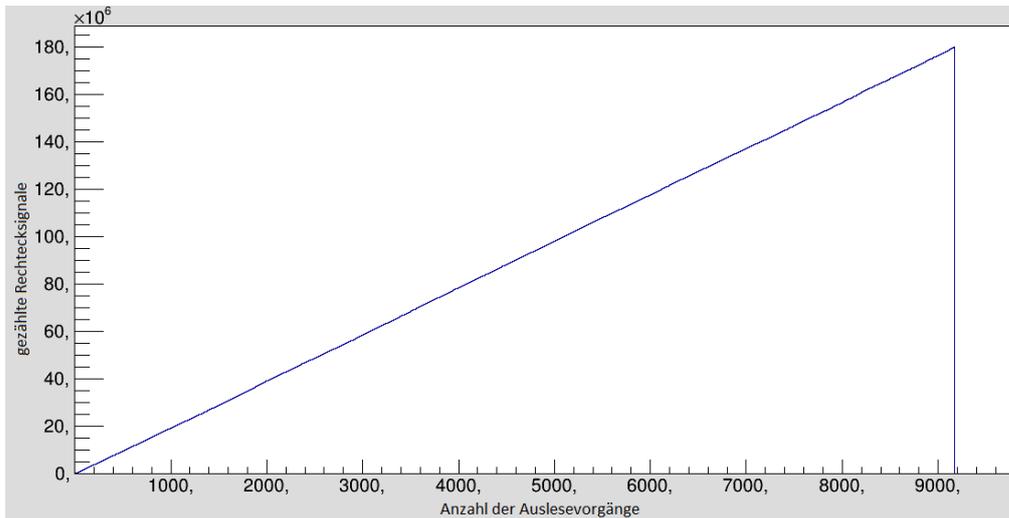


Abbildung 21: Aufgetragen sind die vom Scaler registrierten Rechtecksignale über die Anzahl der Auslesevorgänge. Die Gerade entspricht dem erwarteten Verlauf für einen konstanten Strom.

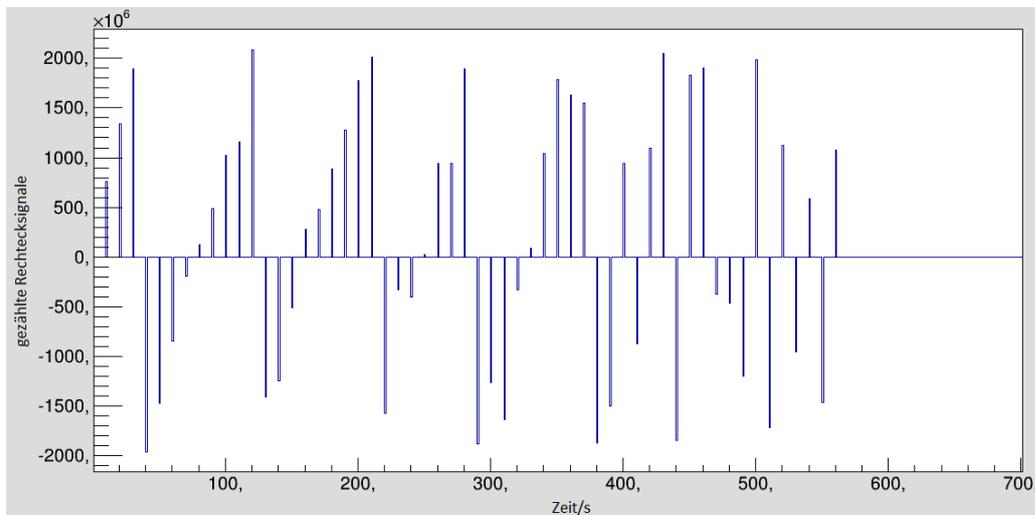


Abbildung 22: Zu sehen sind die gezählten Rechtecksignale in Abhängigkeit von der Zeit. Zu erwarten ist bei einem konstanten Strom jedoch eine Gerade.

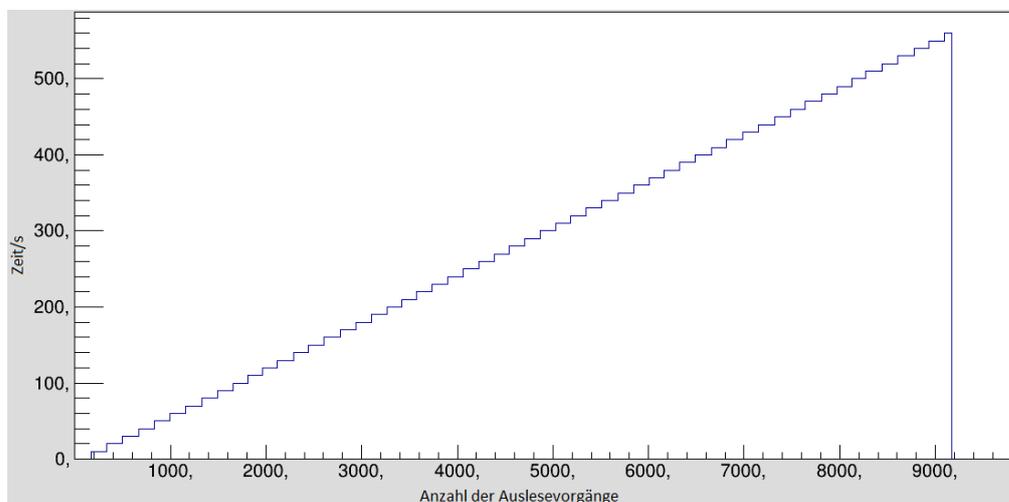


Abbildung 23: Aufgetragen ist der Zeitstempel über die Anzahl der Auslesevorgänge. Der Zeitstempel ist, wie zu sehen ist, über mehrere Auslesevorgänge konstant.

Ein sinnvolles Ergebnis erhält man, indem der Wert des ersten Auslesevorgangs in jedem Intervall konstanter Zeitstempel, über die Zeitstempel aufgetragen wird. Zu sehen ist dies in Abbildung 24. Zwischen den Zeitstempeln bei 10,032 s und 560,644 s wurden 178187254 Signale registriert. Umgerechnet entspricht dies einem Strahlstrom von 985,4 nA. Die Abweichung von dem, vom operator interface, angezeigten Strom beträgt 0,2 %.

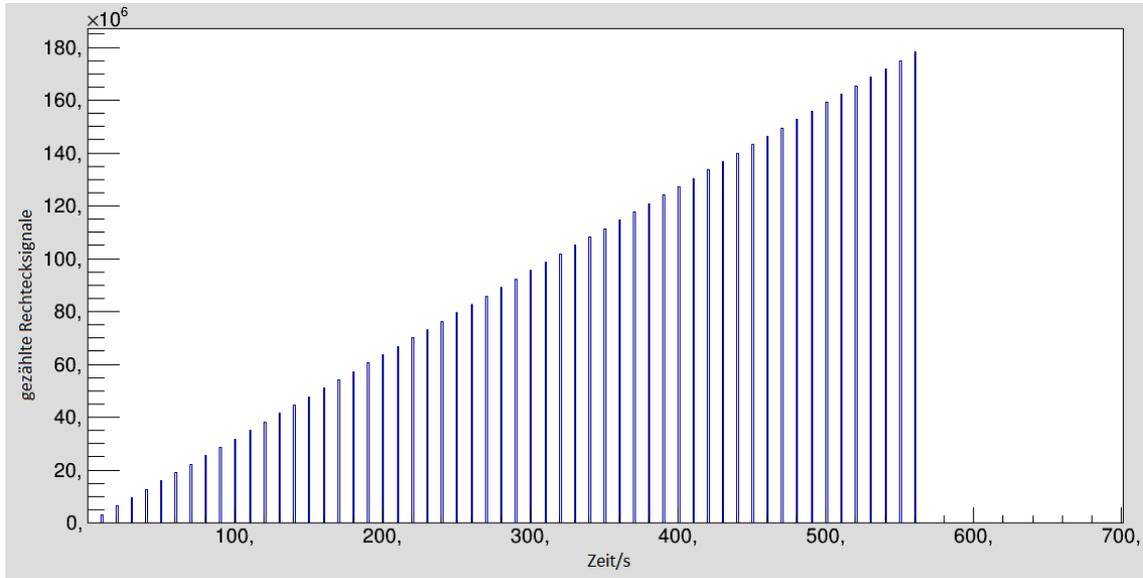


Abbildung 24: Um ein Sinnvolles Ergebnis zu erhalten müssen die Werte des ersten Auslesevorgangs in einem Intervall konstanter Zeitstempel über die Zeitstempel aufgetragen werden.

Durch das oben beschriebene Vorgehen ist es möglich, die gemessene Stromstärke zu rekonstruieren. Die nicht berücksichtigten Werte zwischen zwei Sprüngen der Zeitstempel haben keinen Einfluss auf den berechneten Strom. Allerdings ist es nicht möglich den Strom vor dem ersten bzw. nach dem letzten Sprung zu bestimmen, da für diese Werte keine Anfangs- bzw. Endzeit angegeben werden kann. Dadurch werden bei einer Messung maximal 20 Sekunden unbrauchbar. Für längere Messungen sind diese 20 Sekunden vernachlässigbar.

Um das Problem mit der Zeitmessung zu umgehen, könnte zusätzlich ein Pulser mit konstanter Frequenz an den Scaler angeschlossen werden. Wenn die gezählten Rechtecksignale von Pulser und QM07-Elektronik gleichzeitig gespeichert werden, können die Signale vom Pulser in eine Zeit umgerechnet werden.

3.4. Unsicherheiten der Strom- und Ladungsmessung

Wie in [18] gezeigt werden konnte, sind die Unsicherheiten des gemittelten Stroms I und der Ladung Q durch die Gleichungen (3.6) und (3.7) gegeben.

$$\Delta I_{\text{mittel}} = \sqrt{2 \cdot \left(\frac{\Delta I}{N_{\text{Ende}} - N_{\text{Start}}} \right)^2} \quad (3.6)$$

$$\Delta Q = \sqrt{2 \cdot (I_{\text{mittel}} \cdot \Delta T)^2 + 2 \cdot \left(\frac{T_{\text{Ende}} - T_{\text{Start}}}{N_{\text{Ende}} - N_{\text{Start}}} \cdot \Delta I \right)^2} \quad (3.7)$$

Hierbei sind I der Strom am Faraday-Cup, N die Anzahl der internen Messzyklen der QM07-Elektronik und T die Gesamtmessdauer. Weiterhin konnte gezeigt werden, dass die Unsicherheiten von Strom und Ladung vernachlässigbar sind [18].

Eine Überprüfung der Zuleitung vom Faraday-Cup hinter dem QCLAM-Spektrometer zur QM07-Elektronik, wie sie beim Faraday-Cup hinter dem Lintott-Spektrometer durchgeführt wurde [18], war nicht möglich. Dies liegt daran, dass der Faraday-Cup hinter dem QCLAM-Spektrometer eingemauert und somit nicht zugänglich ist.

Für die Unsicherheit des Strahlstroms, der mit den Messdaten gespeichert werden soll, ist der Auslesevorgang des Scalers von Bedeutung. Während des Auslesevorgangs, der weniger als $1 \mu\text{s}$ dauert [28], wird der Scaler gesperrt. Die Frequenz eines Frequenzgewandelten Strahlstromes von $1 \mu\text{A}$ liegt bei ca. $0,3 \text{ MHz}$. Bei diesem Strahlstrom wird höchstens ein Signal pro Auslesevorgang nicht gezählt. Da aber ca. $0,3 \cdot 10^6$ Signale pro Sekunde ankommen ist der dadurch entstehende Fehler vernachlässigbar, solange der Scaler nicht so oft ausgelesen wird, dass eine nicht vernachlässigbare Anzahl an Rechtecksignalen des frequenzgewandelten Strom registriert wird. Gemessen wurde eine Abweichung von $0,2 \%$, sie kann nicht durch die Sperrung des Scalers verursacht sein. Eine mögliche Ursache könnte in der Problematik der Zeitmessung liegen.

4. Fazit und Ausblick

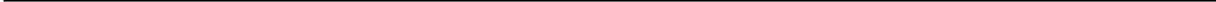
Mit dem neuen Steuerkasten wird es in Zukunft möglich sein, Streuwinkel reproduzierbar anzufahren und die Betriebssicherheit beim Drehen des QCLAM-Spektrometers zu erhöhen. Die Überarbeitung und Erweiterung der Stromauslese der Faraday-Cups hinter den Spektrometern erleichtert die Strahloptimierung sowie die Überwachung des Strahls, während Experimente stattfinden. Zudem kann der gemessene Strahlstrom während des Experimentierbetriebs zusammen mit den Messdaten des QCLAM-Spektrometers gespeichert werden.

Der neue Steuerkasten konnte wegen Vakuumtests am QCLAM-Spektrometer und dem laufendem Strahlbetrieb zeitlich nicht im Rahmen dieser Bachelor-Arbeit getestet werden. Tests mit einem Testaufbau haben gezeigt, dass Drehgeber und Speicherprogrammierbare Steuerung wie vorgesehen funktionieren. Sollte die Spannung des neuen Steuerkastens nicht für das Ansteuern des Frequenzumrichters ausreichen, so wird dieser durch einen neueren ersetzt werden müssen. Da es sich aber um kommerziell erhältliche Bauteile handelt ist der Austausch möglich, ohne die Funktionsfähigkeit zu beeinträchtigen. Nach erfolgreichem Einbau des Steuerkastens wird eine Kalibrierung durchgeführt werden müssen. Um den Steuerkasten vor Strahlenschäden zu schützen ist es sinnvoll einen Stecker einzubauen, damit der Steuerkasten während der Strahlzeit ausgebaut und geschützt gelagert werden kann.

Die neu entwickelte Steuerung kann auch als Vorlage für eine neue Verfahrssteuerung für das Lintott-Spektrometer verwendet werden, da der Aufbau der beiden Spektrometer ähnlich ist.

Durch die Überarbeitung und Erweiterung der Stromauslese mit EPICS ist die Strahloptimierung für das QCLAM-Spektrometer vereinfacht worden. Dies wurde bereits erfolgreich für einen Test des Silizium-Balls in der Strahlzeit Herbst 2013 für einige Stunden verwendet. Die neue Benutzeroberfläche bietet mehr Informationen und vereinfacht durch einen optischen Alarm bei zu geringem Strahlstrom die Überwachung des Experiments. Da es nun möglich ist, den gemittelten Strahlstrom zu speichern, kann auch nach Abschluss der Experimente nach Schwankungen des Strahlstroms gesucht werden. Sollte die Datenaufnahme am Lintott-Spektrometer erneuert werden, so kann die Messung der Ladung ebenfalls mit EPICS erfolgen.

Die Stromauslese ermöglicht es, die während Experimenten am QCLAM-Spektrometer gesammelte Ladung zusammen mit den restlichen Messwerten zu speichern. Um das Problem mit den konstanten Zeitstempeln zu umgehen, kann zusätzlich ein Pulser dem Scaler angeschlossen werden. Mit Bekannter Frequenz des Pulsers ist eine Zeitmessung unabhängig vom Zeitstempel möglich und kann zudem mit höherer Genauigkeit erfolgen. Prinzipiell wurde jedoch die Funktionsfähigkeit des Setups gezeigt. Dieser sollte bei der nächsten Messung am QCLAM-Spektrometer im Experimentierbetrieb getestet werden.



5. Anhang

5.1. easySoft 6

5.1.1. Einleitung

Das Programm easySoft 6 [31] dient zum Erstellen von Steuerprogrammen für speicherprogrammierbare Steuerungen (SPS) von Eaton und ermöglicht das Testen und Übertragen des Steuerprogramms auf das Gerät. Für diese Arbeit wurde das Multifunktionsdisplay MFD-Titan verwendet, bei Geräten anderer Produktreihen kann die Bedienung von easySoft 6 von dieser Beschreibung abweichen. Zum Erstellen, Bearbeiten und Testen des Steuerprogramms ist die kostenlose Demo-Version ausreichend, für die Übertragung des Steuerprogramms auf das Gerät wird jedoch die Vollversion benötigt. Eine ausführlichere Anleitung ist im Beschleunigerwiki des Instituts für Kernphysik der TU Darmstadt [27] zu finden.

5.1.2. Bedienoberflächen

Die fünf (in der Demo-Version vier) wichtigsten Oberflächen sind „Projekt“, „Schaltplan“, „Visualisierung“, „Simulation“ und „Kommunikation“ (fehlt in der Demo-Version). Sie sind ähnlich aufgebaut und bestehen aus drei Feldern. Inhalt und Funktion der Felder hängt von der aktuellen Oberfläche ab. Die Schaltflächen in der Ecke links unten ermöglichen eine Navigation zu der gewünschten Oberfläche. Die Abbildungen Abbildung 25 bis Abbildung 29 zeigen Bildschirm-ausschnitte der Oberflächen.

Der erste Schritt bei der Erstellung eines neuen Steuerprogramms besteht aus der Auswahl der verwendeten Komponenten, dies geschieht in der Oberfläche „Projekt“.

Nach der Auswahl der verwendeten Komponenten kann zu der Oberfläche „Schaltplan“ gewechselt werden. Diese Oberfläche ermöglicht die Erstellung des Steuerprogramms. Das linke Feld bietet nun eine Auflistung der Schaltplanelemente, welche in das obere Feld - das Schaltplanfeld - gezogen werden können

Die Oberfläche „Visualisierung“ dient zum Erstellen von Masken, die auf dem Display angezeigt werden können. Ebenfalls in dieser Oberfläche können die Funktionen der, ins Display integrierten, Tasten festgelegt werden.

Zum Testen von Steuerprogrammen dient die Oberfläche „Simulation“. Während der laufenden Simulation können alle verwendeten Elemente überwacht werden. Dadurch wird das Finden von Fehlern im erstellten Programm erleichtert.

Die Oberfläche „Kommunikation“ dient zur Übertragung des Steuerprogramms auf das Gerät. Um das Gerät mit dem Computer zu verbinden wird ein spezielles USB-Kabel benötigt. Führt das Gerät das Steuerprogramm aus, während es mit dem Computer verbunden ist, so kann am Computer die Funktionsweise des Programms, ähnlich dem Simulationsmodus, überwacht werden.

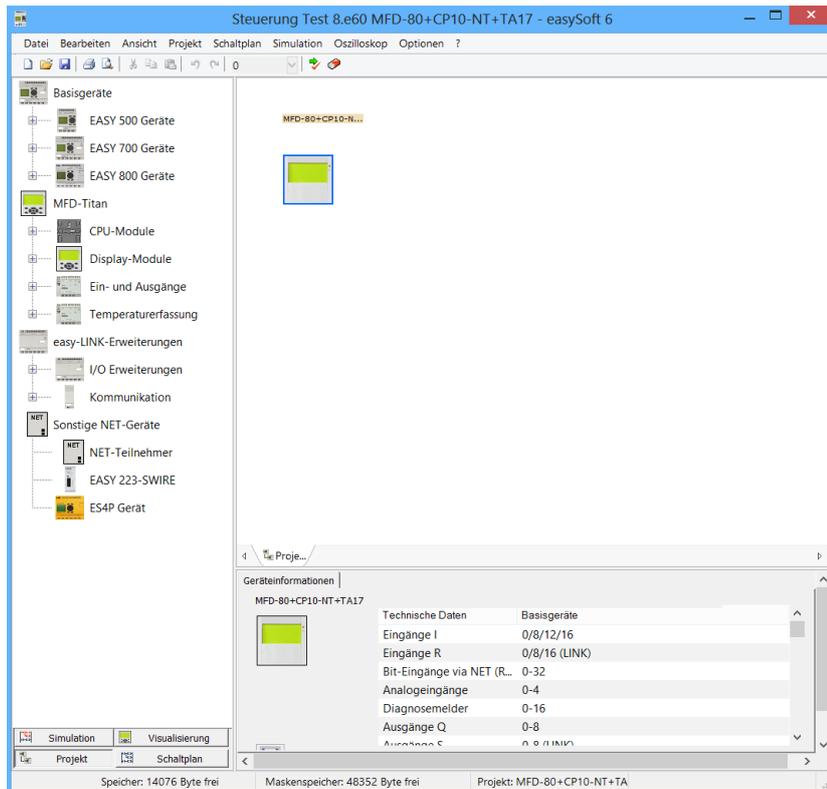


Abbildung 25: Abgebildet ist die Oberfläche „Projekt“, in dieser wird das verwendete Gerät ausgewählt. In der linken Spalte sind die, zur Auswahl stehenden, Geräte aufgelistet. Einige Geräte müssen gegebenenfalls aus mehreren Modulen zusammengesetzt werden.

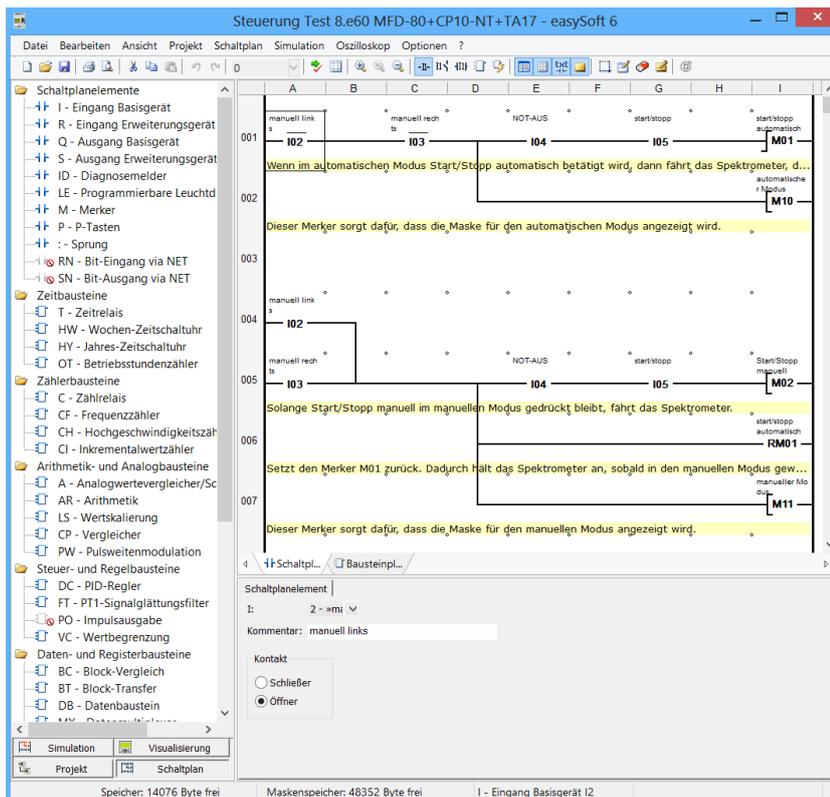


Abbildung 26: Die Oberfläche „Schaltplan“ dient zum Erstellen des Steuerprogramms. Im Schaltplanfeld können die verwendeten Schaltplanelemente angeordnet und verdrahtet werden. Zudem ist es möglich Änderungen an der Funktionsweise der Schaltplanelemente vorzunehmen.

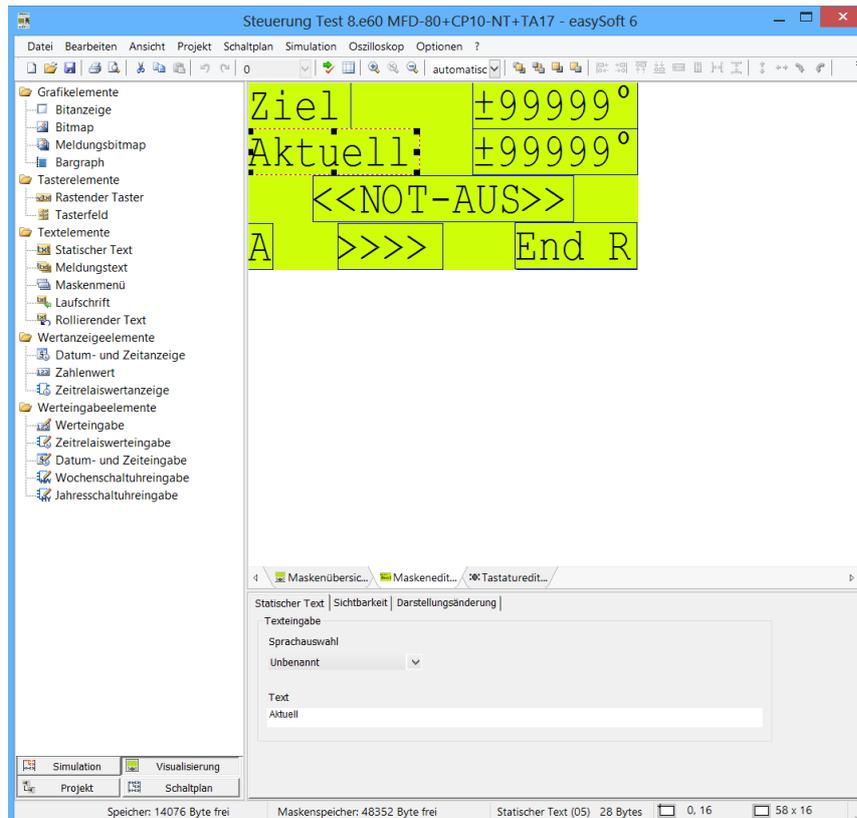


Abbildung 27: Zum Erstellen und Bearbeiten von Masken, sowie zum Festlegen der Funktionen der, ins Display integrierten, Tasten, dient die Oberfläche „Visualisierung“.

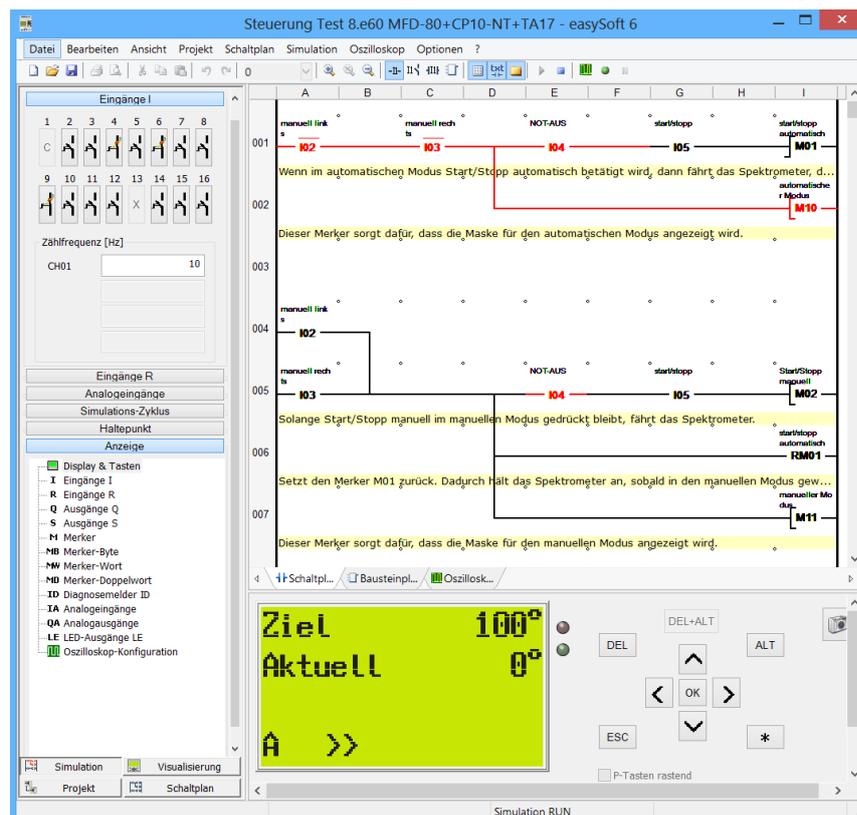


Abbildung 28: Das erstellte Steuerprogramm kann mit der Oberfläche „Simulation“ getestet und auf Fehler überprüft werden.

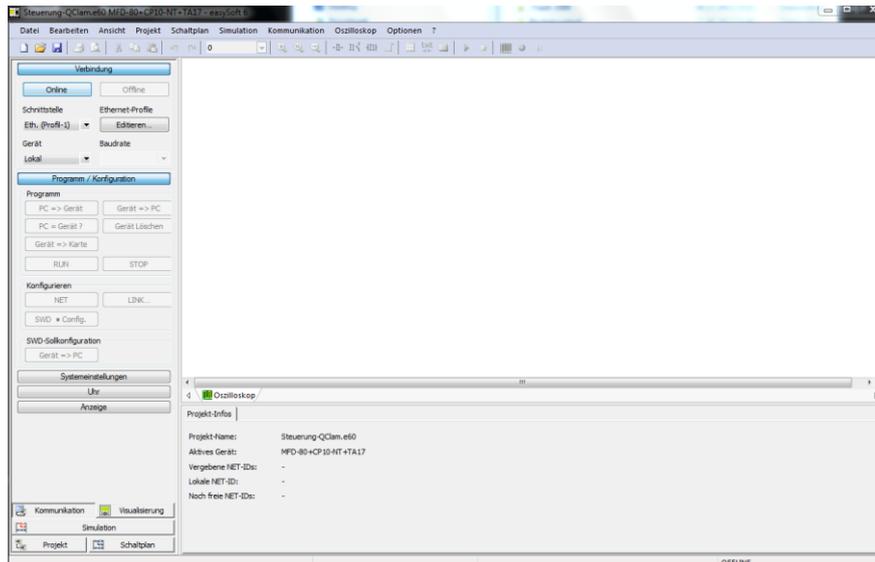


Abbildung 29: Das fertige Steuerprogramm kann mit der Oberfläche „Kommunikation“ über einen USB-Adapter auf das Gerät übertragen werden. Diese Oberfläche ist jedoch nur in der kostenpflichtigen Vollversion von easySoft enthalten.

5.2. Beschreibung der verwendeten Schaltplanelemente.

In diesem Abschnitt werden die verwendeten Schaltplanelemente sowie deren Einstellungsmöglichkeiten beschrieben.

Kontakte:

Einige Schaltplanelemente können als Öffner oder als Schließer verwendet werden. Ein Strich über einem Schaltplanelement zeigt an, dass das Element als Schließer verwendet wird. Es ist zu beachten, dass die Eingänge in der SPS intern als Öffner und Schließer verwendet werden können und unabhängig von den Kontakten der Knöpfe des Steuerkastens sind.

- Öffner: Ein Öffner steht für einen geschlossenen Stromkreis. Liegt an einem Öffner ein Signal an, so wird der Stromkreis geöffnet.
- Schließer: Ein Schließer steht für einen geöffneten Stromkreis. Durch ein anliegendes Signal wird der Stromkreis geschlossen.

Spulenfunktionen:

Bei einigen Schaltplanelementen kann eine Spulenfunktion gewählt werden, diese beeinflusst die Wirkungsweise des Schaltplanelements.

- Rücksetzen: Das Schaltplanelement wird auf den Standardwert zurückgesetzt.
- Schütz: Solange ein Signal anliegt wird auch eines weitergegeben.
- Stromstoß: Durch ein einmaliges Signal wird das Schaltplanelement gesetzt, ein erneutes Signal setzt das Schaltplanelement zurück.

Übersicht Strompfade:

Mit Strompfaden können die Schaltplanelemente verbunden werden. Sie werden mit schwarzen Strichen dargestellt.

Übersicht Eingänge:

Eingänge registrieren Signale an den Eingängen der SPS. Analogeingänge messen die Spannungen im Intervall 0V bis 10V mit einer Auflösung von 1024 Schritten.

- I01: Ist hardwaremäßig mit dem Hochgeschwindigkeitszähler verbunden und darf daher für nichts anderes verwendet werden. Angeschlossen ist der Drehgeber.
- I02: Manuell links (entspricht der Richtung zu größeren Streuwinkeln)
- I03: Manuell rechts (entspricht der Richtung zu kleineren Streuwinkeln)
- I04: NOT-AUS
- I05: Start/Stopp
- I06: Endschalter links
- I07: Reset
- I09: Endschalter rechts
- I10: Geschwindigkeit
- I12: Dieser Analogeingang ist mit dem Wendepotentiometer verbunden.

Übersicht Ausgänge:

Ausgänge steuern die Signale an den Ausgängen der SPS. Analogausgänge geben Spannungen von 0 V bis 10 V in 1024 Schritten aus.

- Q1: Nach links fahren
- Q2: Nach rechts fahren
- QA1: Geschwindigkeit

Übersicht Arithmetikbausteine:

Mit Arithmetikbausteinen können simple Berechnungen wie Addition, Subtraktion, Multiplikation und Division durchgeführt werden.

- AR01: Potentiometer einlesen 1
- AR02: Potentiometer umrechnen
- AR03: Potentiometer einlesen 2
- AR04:
- AR05: Zähler einlesen
- AR06: untere Grenze
- AR07: obere Grenze
- AR08: langsam fahren einlesen
- AR09: schnell fahren einlesen
- AR10: Analogausgang

Übersicht Zeitrelais:

Zeitrelais ermöglichen es Signale zu verzögern beziehungsweise um eine vorgegebene Zeit zu verlängern.

- T01: Ausrollen
- T02: verzögertes losfahren

Übersicht Vergleicher:

Vergleicher vergleichen zwei Werte und geben, abhängig von ihrer Einstellung, ein Signal aus, wenn der erste Wert größer beziehungsweise kleiner als der zweite Wert ist.

- CP01: aktuell kleiner als eingestellt
- CP02: aktuell größer als eingestellt

Übersicht LED:

Es stehen eine grüne (LED3) und eine rote (LED2) LED zur Verfügung, LED1 verdunkelt lediglich das Display.

- LE1: Warnung NOT-AUS
- LE2: Warnung Endschalte/NOT-AUS
- LE3: es wird gefahren

Übersicht Hochgeschwindigkeitszähler:

Hochgeschwindigkeitszähler können Rechtecksignale mit einer Frequenz von bis zu 3000 Hz zählen. Dabei wird die Taktfrequenz des CPU-Moduls umgangen.

- CH01: Zähler

Übersicht Merker:

Merker können zum Speichern von binären Informationen verwendet werden.

- M01: start/stopp automatisch
- M02: start/stopp manuell
- M03: nicht am Ziel 1
- M04: fahren links
- M05: nicht am Ziel 2
- M06: fahren rechts
- M07: langsam/schnell
- M08: Fahrtrichtung links
- M09: Fahrtrichtung rechts
- M10: automatischer Modus
- M11: manueller Modus
- M81: fahren
- M89: schnell fahren

Übersicht Doppelwörter:

Doppelwörter werden zum Speichern von Informationen verwendet. Ein Doppelwort besteht aus zwei Wörtern, welche wiederum aus zwei Byte bestehen.

- Doppelwort01: Die ersten elf Bit im Doppelwort01 sind von den Merkern M01 bis M11 belegt.
- Doppelwort03: Dieses Doppelwort wird für die Berechnung der Geschwindigkeit verwendet.

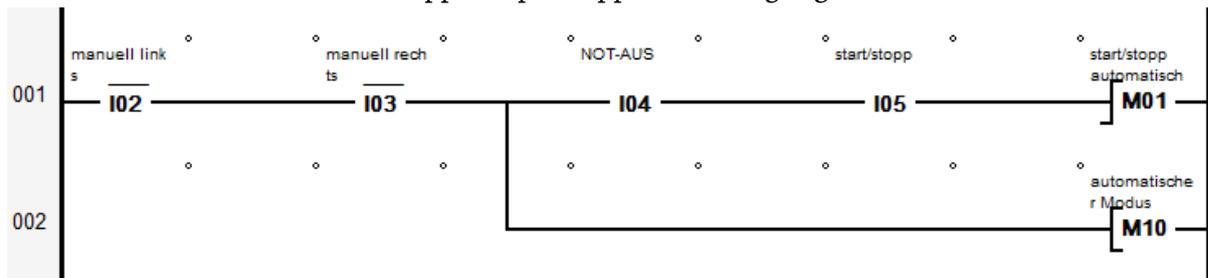
5.3. Steuerprogramm

In diesem Abschnitt wird zuerst das Steuerprogramm erklärt. Abschließend werden die Masken, welche auf dem Display der Speicherprogrammierbaren Steuerung zu sehen sind, beschrieben.

Zeilen 1-2:

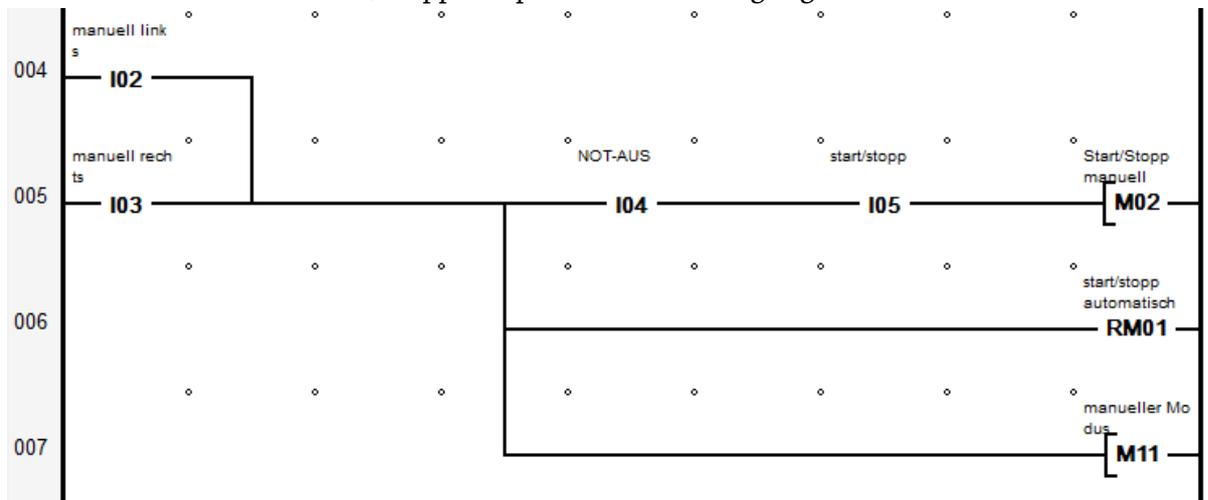
Wenn an den Eingängen I02 (manuell links) und I03 (manuell rechts) keine Signale anliegen befindet sich die Steuerung im automatischen Modus. Der Merker M10 (Anzeige automatischer Modus) sorgt für die Anzeige der Maske „automatischer Modus“ auf dem Display. Wird nun, bei nicht ausgelöstem Not-Aus-Taster, der Knopf start/stopp einmal

gedrückt, dann wird das Signal für fahren durch den Merker M01 weitergegeben. Ein erneuter Druck auf den start/stopp-Knopf stoppt die Bewegung.



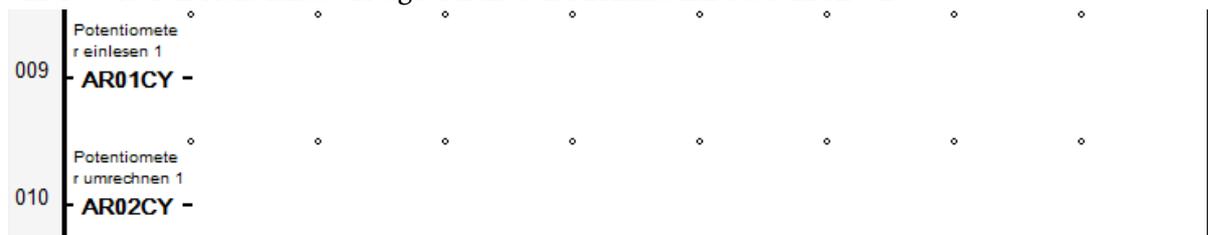
Zeilen 4-7:

Bei manueller Vorgabe einer Fahrtrichtung (an I02 oder I03 liegt ein Signal an) befindet sich die Steuerung im manuellen Modus. Durch Rücksetzen des Merkers M01 wird die Bewegung gestoppt, falls dies nicht bereits geschehen ist. Der Merker M11 sorgt für die Anzeige der Maske des manuellen Modus auf dem Display. In diesem Modus wird der Befehl „fahren“ bei nicht ausgelöstem Not-Aus-Taster weitergegeben solange der start/stopp-Knopf gedrückt wird. Ein Loslassen des start/stopp-Knopfes hält die Bewegung sofort an.



Zeilen 9-10:

Mit den beiden Arithmetikbausteinen wird die, am Analogeingang IA4 anliegende, Spannung eingelesen und mit einer Geradengleichung in den Streuwinkel umgerechnet. Der Arithmetikbaustein AR01 übernimmt die Multiplikation mit einer Konstanten, die Addition einer weiteren Konstante erfolgt durch den Arithmetikbaustein AR02.



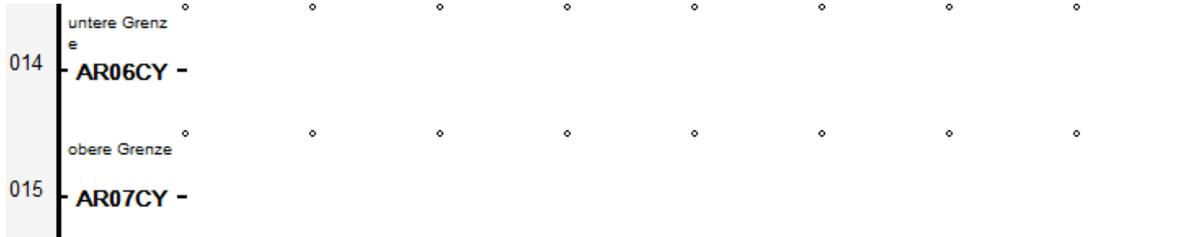
Zeile 12:

Der Wert vom Analogeingang IA4 wird erneut eingelesen, diesmal jedoch nur mit einer Konstanten multipliziert. Dieser Wert wird für den Reset des Hochgeschwindigkeitszählers verwendet (siehe Zeile 40).



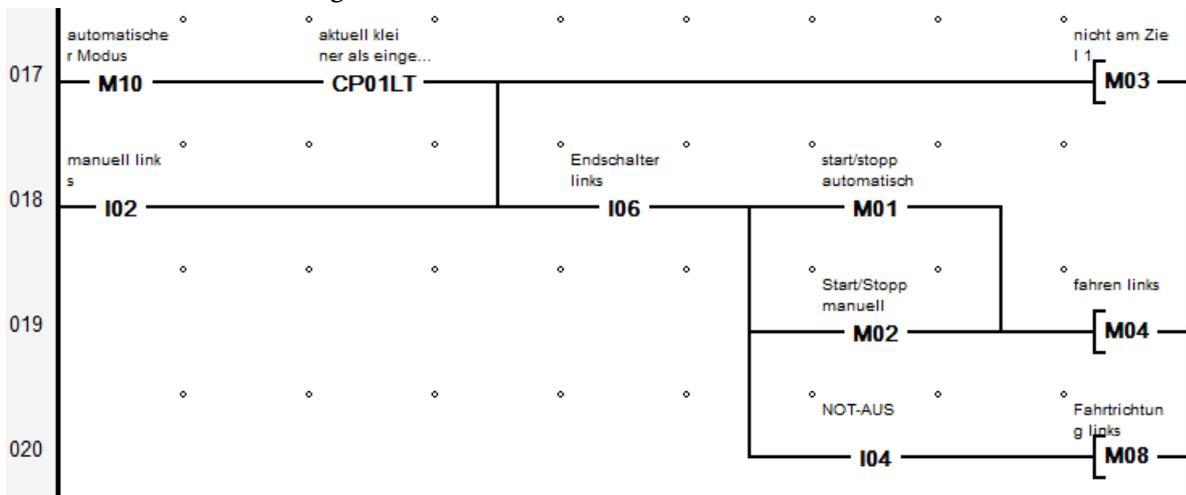
Zeilen 14-15:

Die beiden Arithmetikbausteine berechnen eine obere beziehungsweise untere Grenze des vorgegebenen Streuwinkels. Dadurch wird ein kleiner Toleranzbereich um den eingestellten Winkel gebildet, welcher im automatischen Modus den Zielbereich darstellt. Dadurch soll verhindert werden, dass aufgrund der hohen Zählrate des Hochgeschwindigkeitszählers am Ziel vorbeigefahren wird.



Zeilen 17-20:

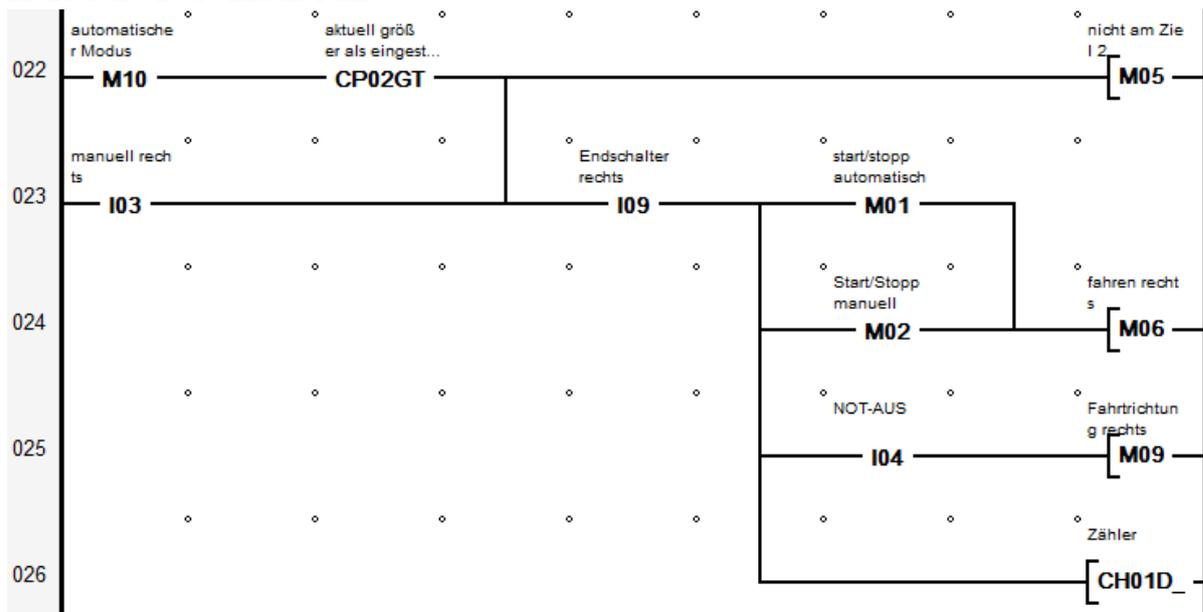
In diesen vier Zeilen werden die Fahrtrichtung links (Merker M08) festgelegt und das Signal für die Fahrtrichtung weitergegeben (Merker M04), sofern dies nicht in diesen Zeilen verhindert wird. Dies kann wegen einer Besonderheit des automatischen Modus nicht mit einem einzigen Merker realisiert werden (siehe Zeilen 47-51). Manuell wird die Fahrtrichtung links durch ein Signal an I02 eingestellt. Im automatischen Modus vergleicht der Vergleichler CP01 den aktuellen Winkel (siehe Zeile 45) mit der unteren Grenze des Arithmetikbausteins AR06. Falls der aktuelle Winkel kleiner als die untere Grenze ist entspricht dies der Fahrtrichtung links. Ist dies erfüllt registriert der Merker M03, dass das Ziel in dieser Fahrtrichtung noch nicht erreicht wurde. Diese Fahrtrichtung wird jedoch blockiert, sobald der linke Endschalter ausgelöst wird.



Zeilen 22-26:

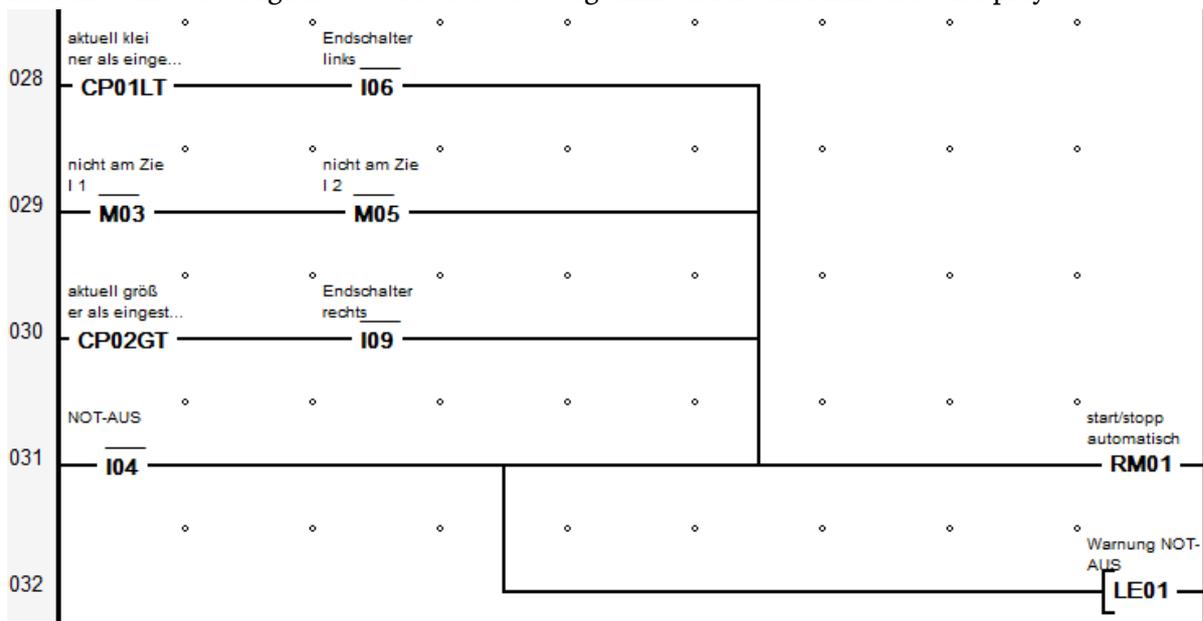
Dieser Teil dient für die Fahrtrichtung rechts und ist analog zu den Zeilen 17-20 aufgebaut. Der Vergleichler CP02 überprüft, ob der aktuelle Winkel größer als die obere Grenze von AR07 ist. Die Merker M05, M06 und M09 entsprechen den Merkern M03, M04 und M08. Ein Unterschied zwischen den Zeilen 17-20 und 22-26 ist in Zeile 26 zu sehen, dort wird die

Zählrichtung des Hochgeschwindigkeitzählers geändert, da die Fahrtrichtung rechts zu kleineren Streuwinkeln führt.



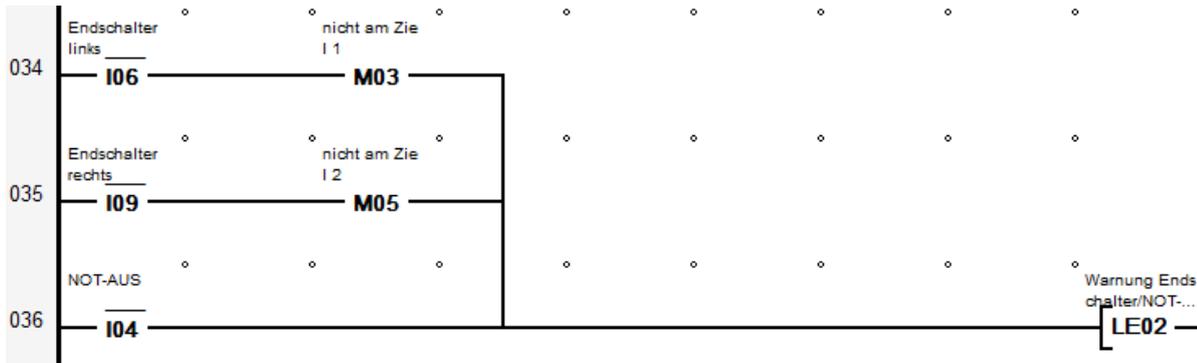
Zeilen 28-32:

Hier werden verschiedene Ursachen für einen automatischen Stopp im automatischen Modus festgelegt, das heißt der Merker M01 wird zurückgesetzt. Dazu gehört das Erreichen des Zielintervalls. Dies geschieht, wenn die Merker M03 und M05 nicht gesetzt sind, d. h. wenn der aktuelle Winkel zwischen der unteren und der oberen Grenze liegt. Aber auch eine Betätigung des Not-Aus-Tasters oder des Endschalters in Fahrtrichtung setzen M01 zurück. Im Falle eines betätigten Not-Aus-Tasters sorgt LED01 für ein blinkendes Display.



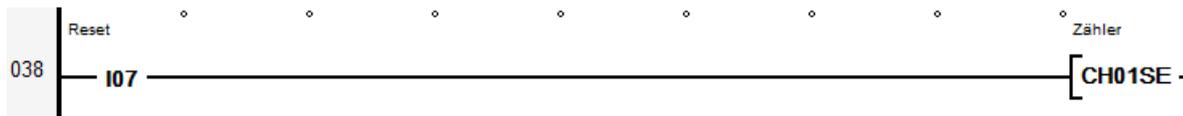
Zeilen 34-36:

Die rote Leuchtdiode rechts neben dem Display leuchtet wenn der Endschalter in Fahrtrichtung oder der Not-Aus-Taster betätigt werden.



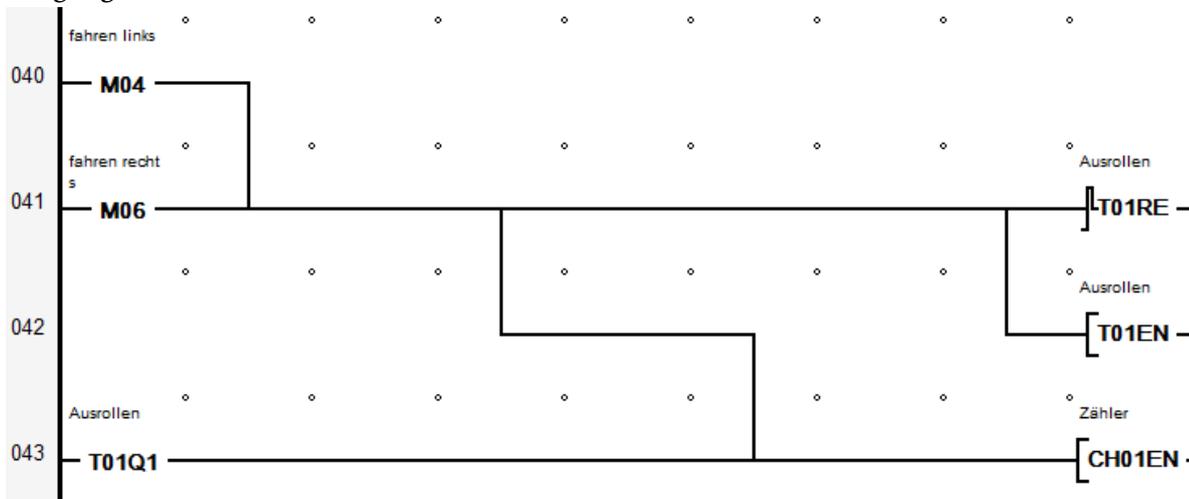
Zeile 38:

Ein Signal am Eingang I07 setzt den Wert des Hochgeschwindigkeitszählers auf den Wert von AR03.



Zeilen 40-43:

Während das Spektrometer fährt (M04 oder M06 ist gesetzt) wird der Hochgeschwindigkeitszähler CH01 freigegeben und zählt am Eingang I1 ankommende Rechtecksignale. Das Zeitrelais T01 startet nach dem Stopp eine Verzögerung von drei Sekunden. In dieser Zeitspanne ist der Hochgeschwindigkeitszähler weiterhin aktiv und kann alle Signale bis zum völligen Stillstand zählen. Danach kann auch am Drehgeber gearbeitet werden ohne den Streuwinkel zu verfälschen. Sollte innerhalb dieser drei Sekunden erneut gefahren werden, dann wird das Zeitrelais zurückgesetzt und startet nach dem nächsten Stopp die Verzögerung erneut. Da das Programm innerhalb einer Zeile von links nach rechts und die Zeilen von oben nach unten abgearbeitet werden [14], kann sich das Zeitrelais durch seinen Ausgang selbst zurücksetzen.



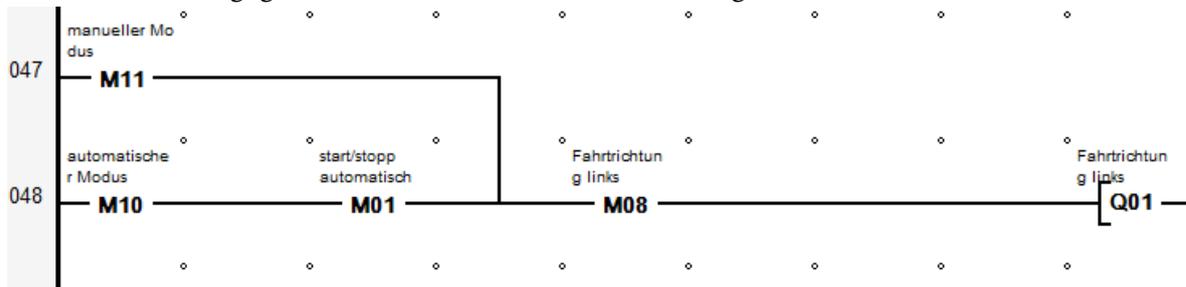
Zeile 45:

Der Arithmetikbaustein AR05 nimmt den Wert vom Hochgeschwindigkeitszähler und multipliziert ihn mit einer Konstante (die Addition findet bereits beim Reset statt). Das Ergebnis ist der aktuelle Winkel.



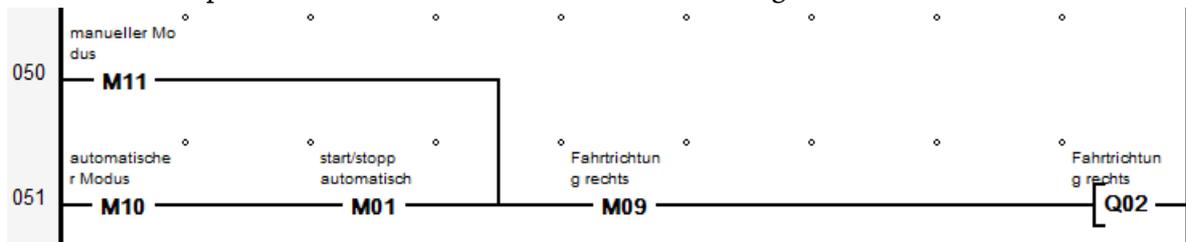
Zeilen 47-48:

Im manuellen Modus wird, wenn M08 gesetzt ist, die Fahrtrichtung links an Q01 ausgegeben. Befindet sich die Steuerung im automatischen Modus muss zusätzlich der Merker M01 gesetzt sein. Dadurch wird die Richtung im automatischen Modus, erst bei einem Druck auf den „Start/Stop“ Knopf festgelegt. Somit wird verhindert, dass Spannungsschwankungen am Analogeingang IA04 zu mehreren Richtungswechseln pro Sekunde führen können, wenn aktueller und vorgegebener Winkel nahe beieinander liegen.



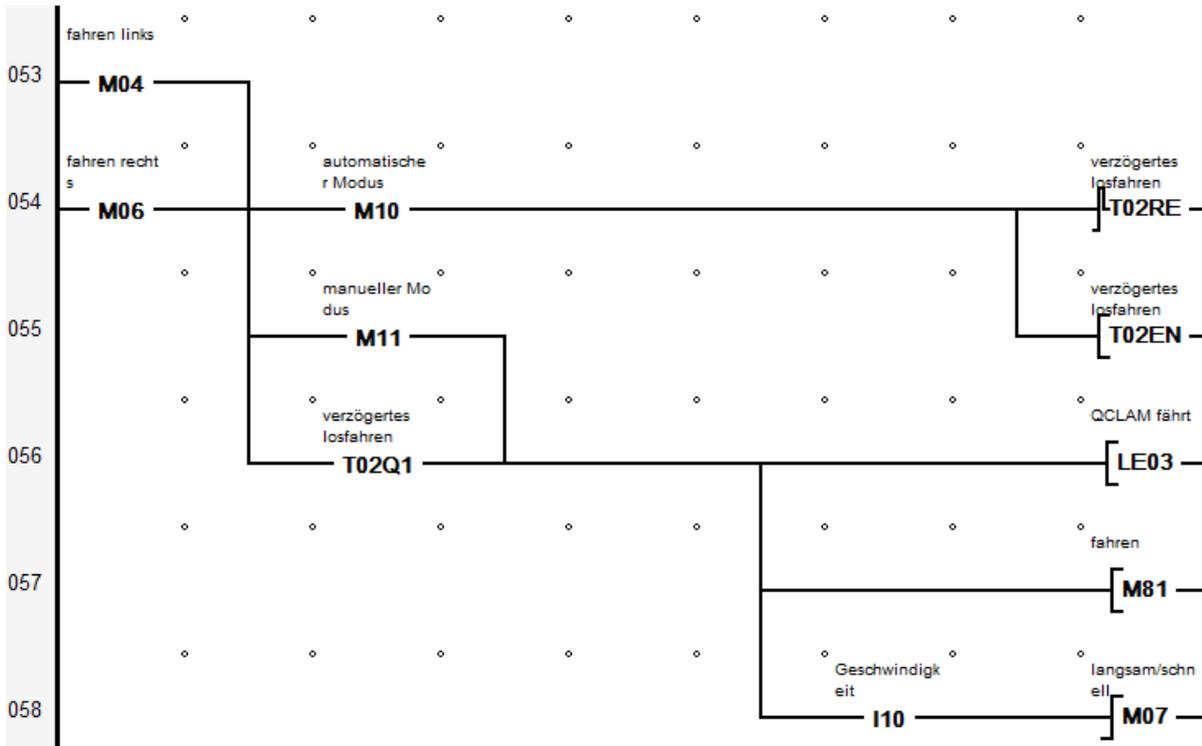
Zeilen 50-51:

Dieser Teil entspricht den Zeilen 47-48 für die Fahrtrichtung rechts.



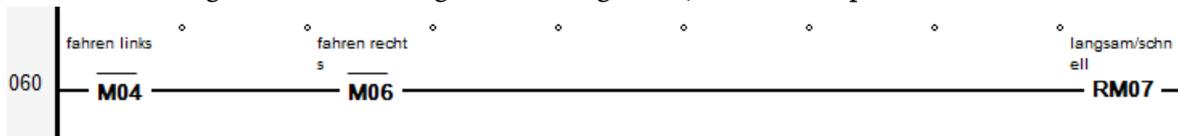
Zeilen 53-58:

Wenn das Spektrometer fahren soll wird zunächst geprüft, ob sich die Steuerung im automatischen Modus befindet. Trifft dies zu wird durch das Zeitrelais T02 ein verzögertes Losfahren bewirkt. Dadurch soll verhindert werden, dass die Auswahl der Fahrtrichtung zeitgleich mit dem Losfahren erfolgt. Diese Verzögerung wird im manuellen Modus umgangen. Während gefahren wird, leuchtet als Hinweis die grüne LED. Mit einem Signal am Eingang I10 kann, während gefahren wird zwischen zwei Geschwindigkeitsstufen gewählt werden. Der Merker stellt das niedrigste Bit im Byte 11 dar und wird für die Geschwindigkeitsregulierung verwendet (siehe Zeile 67).



Zeile 60:

Die Geschwindigkeit wird auf langsam zurückgesetzt, wenn das Spektrometer nicht fährt.



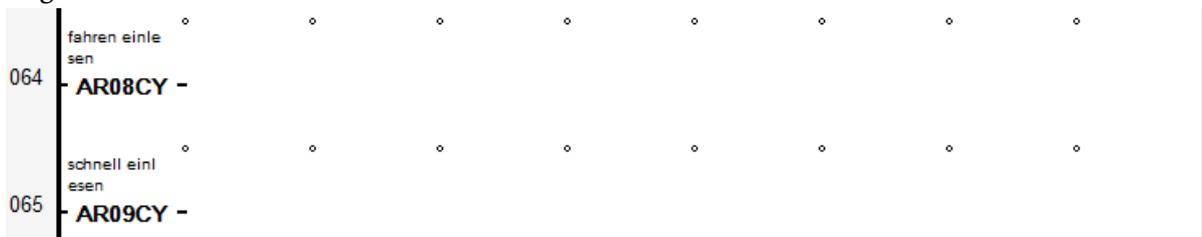
Zeile 62:

Wenn schnell gefahren werden soll wird der Merker 89 gesetzt. Dieser Merker stellt das niedrigste Bit im Byte 12 dar und wird für die Geschwindigkeitsregulierung verwendet.



Zeile 64:

Die beiden Arithmetikbausteine führen Berechnungen für die Geschwindigkeitsausgabe am Analogausgang QA1 aus. Während nicht gefahren wird, liefern beide das Ergebnis Null, wird langsam gefahren, liefert AR09 das Ergebnis Null. Bei schneller Fahrt sind beide Ergebnisse ungleich Null.



Zeile 67:

Der Arithmetikbaustein AR10 addiert die Werte der Arithmetikbausteine AR08 und AR09. Das Ergebnis wird am Analogausgang QA1 ausgegeben (1023 entspricht 10 V). Über diese Spannung kann die Geschwindigkeit des Motors gesteuert werden



Masken:

Es wurden zwei Masken erstellt. Dabei handelt es sich um eine Maske für den automatischen Modus und eine Maske für den manuellen Modus. Die Merker M10 und M11 sorgen für das Anzeigen der Maske, die zur aktuellen Betriebsart gehört. In beiden Masken werden die Werte der Arithmetikbausteine AR02 (vorgegebener Winkel) und AR05 (aktueller Winkel) angezeigt. Die Sichtbarkeit der Endschalter und des Not-Aus-Tasters ist mit den entsprechenden Eingängen verknüpft, dies gilt auch für die Geschwindigkeitsanzeige. Im manuellen Modus wird die eingestellte Fahrtrichtung durch die Eingänge IO2 beziehungsweise IO3 blinkend hervorgehoben.

5.4. Bedienungsanleitung

Die Steuerung verfügt über zwei Betriebsmöglichkeiten, den automatischen und den manuellen Modus. Zunächst werden die beiden Betriebsarten und ihre Bedienung einzeln vorgestellt, abschließend werden die Gemeinsamkeiten erklärt.

5.4.1. Der automatische Modus

Steht der Drehschalter auf der mittleren Position, so befindet sich die Steuerung im automatischen Modus. Dies wird durch ein „A“ in der Ecke links unten verdeutlicht. Eine Einstellung des Zielwinkels erfolgt durch den Drehregler mit der Beschriftung „Winklereinstellung“, angezeigt wird der Zielwinkel im Display in der obersten Zeile mit der Bezeichnung „Ziel“. Unter dem Zielwinkel wird, mit der Bezeichnung „Aktuell“, die aktuelle Position des Spektrometers angezeigt. Nachdem ein Zielwinkel eingestellt wurde kann durch einmaliges drücken des „Start/Stop“ Knopfes das Spektrometer in Bewegung gesetzt werden. Während das Spektrometer gefahren wird leuchtet rechts neben dem Display eine grüne LED. Die Bewegung wird bei Erreichen des Zielwinkels oder Betätigen des Endschalters in Fahrtrichtung automatisch gestoppt. Manuell kann die Bewegung durch erneutes Drücken des „Start/Stop“ Knopfes oder natürlich durch Betätigen des Not-Aus-Tasters angehalten werden. Wenn ein Endschalter betätigt wird oder der Not-Aus-Taster betätigt wird, leuchtet die rote LED rechts neben dem Display. Wurde das Spektrometer durch einen Endschalter oder den Not-Aus-Taster gestoppt, dann wird im Steuerprogramm der Merker für fahren zurück gesetzt. Dadurch fängt das Spektrometer nach Entsperrung des Not-Aus-Tasters nicht automatisch an zu fahren.

5.4.2. Der manuelle Modus

Im Gegensatz zum automatischen Modus fährt das Spektrometer in die, am Drehschalter eingestellte Richtung und nicht zum vorgegebenen Winkel. Gekennzeichnet wird der manuelle Modus durch ein „M“ in der Ecke links unten. Rechts daneben wird mit „R/L“ die eingestellte Bewegungsrichtung angezeigt, dabei blinkt der, der Bewegungsrichtung

entsprechende, Buchstabe. Eine Drehung des Drehschalters zur linken bzw. rechten Position ändert die Betriebsart von automatisch zu manuell. Der Drehregler dient in dieser Betriebsart lediglich zum Reset des Winkels (siehe Anhang) und hat keine weitere Funktion, daher ist der Winkel im manuellen Modus mit „Vorgabe“ gekennzeichnet. Unterhalb der „Vorgabe“ wird weiterhin mit der Kennzeichnung „Aktuell“ die aktuelle Position angezeigt. Um das Spektrometer zu fahren, muss der Knopf „Start/Stopp“ gedrückt gehalten werden. So lange das Spektrometer gefahren wird, leuchtet die grüne LED rechts neben dem Display. Wird der Knopf losgelassen, der Endschalter in Fahrtrichtung betätigt oder der Not-Aus-Taster betätigt, dann stoppt die Bewegung.

Gemeinsamkeiten:

In beiden Betriebsarten ist zu Beginn der Bewegung die Geschwindigkeit auf langsam gestellt. Ein Umschalten zwischen den Geschwindigkeitsstufen langsam und schnell erfolgt nur durch drücken des Knopfs „Geschwindigkeit“ während das Spektrometer gefahren wird. Bei einem Stopp des Spektrometers wird die Geschwindigkeitsstufe gleichzeitig auf langsam gesetzt. Der Knopf „Reset“ setzt den aktuellen Winkel auf den vorgegebenen Winkel. Der vorgegebene Winkel wird mit dem Drehregler eingestellt und auf dem Display in der obersten Zeile angezeigt.

5.5. Verwendete Teile

Die verwendeten Teile sind in drei Tabellen aufgelistet. In Tabelle 1 sind Drehgeber, Multifunktionsdisplay und das Netzteil zu finden. Die Gehäuse sind in Tabelle 2 aufgelistet, eine Übersicht über alle verwendeten Schaltelemente ist in Tabelle 3 zu finden

Tabelle 1: Aufgelistet sind der Endschalter, alle Teile des Multifunktionsdisplays und das Netzteil.

Teil	Hersteller	Bezeichnung
Drehgeber	Kübler	8.3620.543E.1000
Positionsschalter	Pizzato Elettrica	FR 531-M2
MFD TITAN CPU-Modul	Eaton	MFD-CP10-NT
MFD TITAN Input/Output Modul	Eaton	MFD-TA17
MFD TITAN Display	Eaton	MFD 80
Netzteil	Eaton	Easy400-POW

Tabelle 2: Aufgelistet sind die beiden Gehäuse und deren Bezeichnungen. Zusammengehalten werden sie mit zwei M8 Schrauben.

Teil	Bezeichnung
Großes Gehäuse	Universal-Gehäuse ABS Gehäuse 240x160x90 Bopla T 2401
Kleines Gehäuse	Universal-Gehäuse ABS 160x83x52 Strapubox

Tabelle 3: Aufgelistet sind die Schalter sowie die dazugehörigen Adapter und Kontaktelemente. In der rechten Spalte ist die Funktion beschrieben. Zusammengehörige Schalter, Adapter und Kontaktelemente sind in dieser Reihenfolge untereinander angeordnet.

Teil	Anzahl	Hersteller	Bezeichnung	Funktion
Not-Aus-Taster	1	Eaton	Not-Aus-Taster Q25 24V/DC	Not-Aus
Kontaktelemente	2	Eaton	Kontaktelement E01	-
Flacher Drucktaster	3	BACO	Drucktaster L21AA02	Start/Stopp Geschwindigkeitsauswahl Reset
Adapter	3	BACO	Adapterplatte 333E	-
Kontaktelemente	3	BACO	Kontaktelement3 3E10	-
Wahltaste	1	Pizzato Elettrica	Wahltaste rund E21SE E21SE43ACE11AB	Manuell links Manuell rechts Automatisch
Adapter	1	Pizzato Elettrica	E21BAC11	-
Kontaktelemente	2	Pizzato Elettrica	E2CP10G2V1	-
Wendelpotentiometer 2 k Ω (mit $-\Omega$ Widerstand in Reihe geschaltet)	1			Winkeleinstellung

5.6. Stromauslese mit EPICS

Wurden alle Daten, wie in [22] beschrieben, heruntergeladen, dann steht nun ein Ordner mit der Bezeichnung acs bereit. Der input/output controller (IOC) für die Stromauslese des Faraday-Cups hinter dem Lintott-Spektrometer befindet sich in acs/frameworks/epics/iocapps/Lintott. Von diesem Ordner wurde eine Kopie angelegt und in Spektrometer umbenannt. Um diese Umbenennung konsequent fortzusetzen war es wichtig in den folgenden Dateinamen und Ordner lintott durch spektrometer zu ersetzen:

- acs/frameworks/epics/iocapps/Spektrometer/trunk/iocBoot/
lintottApp -> spektrometerApp
- acs/frameworks/epics/iocapps/spektrometer/trunk/iocBoot/spektrometerApp/DB/
lintott.substitutions -> spektrometer.substitutions
- acs/frameworks/epics/iocapps/Spektrometer/trunk/iocBoot/spektrometerApp/src/
lintottMain.cpp -> spektrometerMain.cpp

Auch in den folgenden Dateien musste lintott durch spektrometer ersetzt werden:

- acs/frameworks/epics/iocapps/Spektrometer/trunk/iocBoot/ioclinux-x86/st.cmd

```
#!../bin/linux-x86/spektrometer

## You may have to change spektrometer to something else
## everywhere it appears in this file

< envPaths

cd ${TOP}

## Register all support components
dbLoadDatabase("dbd/spektrometer.dbd",0,0)
spektrometer_registerRecordDeviceDriver(pdbbase)

## Load record instances
dbLoadTemplate "db/spektrometer.substitutions"

devSocketCanInit

cd ${TOP}/iocBoot/${IOC}
iocInit()

## Start any sequence programs
#seq sncspektrometer,"user=burandt"
```

- acs/frameworks/epics/iocapps/Spektrometer/trunk/iocBoot/ioclinux-x86_64/st.cmd

```
#!../bin/linux-x86_64/spektrometer

## You may have to change spektrometer to something else
## everywhere it appears in this file

< envPaths
```

```

cd                                                    ${TOP}

##          Register          all          support          components
dbLoadDatabase("dbd/spektrometer.dbd",0,0)
spektrometer_registerRecordDeviceDriver(pdbbase)

##          Load          record          instances
dbLoadTemplate          "db/spektrometer.substitutions"

devSocketCanInit

cd                                                    ${TOP}/iocBoot/${IOC}
iocInit()

##          Start          any          sequence          programs
#seq sncspektrometer,"user=burandt"

```

Um einen Grenzwert für den Strahlstrom vorgeben zu können wurde das Template `spektrometet_cup_alert.template` erstellt. Dadurch kann im operator interface ein Wert eingegeben werden, ebenfalls im operator interface wird dieser Wert mit dem gemittelten Strahlstrom verglichen.

- `acs/frameworks/epics/iocapps/Spektrometer/trunk/spektrometerApp/Db/spektrometer_cup_alert.template`

```

record(ao, "$(name):alert") {
field(PREC, "1")
field(EGU, "nA")
}

```

In dem Unterordner `acs/frameworks/epics/iocapps/Spektrometer/trunk/spektrometerApp/Db` mussten die Dateien `spektrometer.substitutions` und `Makefile` bearbeitet werden, da noch das veraltete Template `qm07channel.template` verwendet wurde und weitere Funktionen hinzugefügt werden sollten.

Für die Stromauslese wird nun das `qm07_iadc.template.pre` verwendet. Neu hinzugefügt wurden `qm07_iadc_sum.template.pre` und `qm07avgChannel.template`. Das Template `qm07_iadc_sum.template.pre` ermöglicht es, die von den Faraday-Cups aufgefangene Ladung auszulesen. Um den Strom in der Beschleunigerdatenbank zu speichern, aber nicht zu viel Speicherplatz zu belegen wurde, das Template `qm07avgChannel.template` verwendet. Dadurch wird der Strom über eine Sekunde gemittelt und überprüft, ob sich dieser Wert von dem vorherigen um mehr als einen vorgegebenen Wert unterscheidet. Somit wird das Rauschen unterdrückt und die zu speichernde Datenmenge kann reduziert werden. Auch das Template `spektrometer_cup_alert.template` muss in die Datei `Makefile` integriert werden. Die Verweise zu den vier Templates wurden in der Datei `Makefile` analog zu den vorhandenen Verweisen erstellt.

- `acs/frameworks/epics/iocapps/Spektrometer/trunk/spektrometerApp/Db/Makefile`

```

TOP=../../
include                                                    ${TOP}/configure/CONFIG

```

```

#-----
#          ADD          MACRO          DEFINITIONS          AFTER          THIS          LINE
#-----
#          Optimization  of  db  files  using  dbst  (DEFAULT:  NO)
#DB_OPT          =          YES
#-----
#          Create  and  install  (or  just  install)  into  <top>/db
#          databases,  templates,  substitutions  like  this
DB          +=          can_ifc.template
DB          +=          qm07_iadc.template
DB          +=          qm07_iadc_sum.template
DB          +=          qm07crate.template
DB          +=          qm07card.template
DB += qm07_avgChannel.template
DB          +=          spektrometer_cup_alert.template
DB          +=          spektrometer.substitutions
#-----
#          If  <anyname>.db  template  is  not  named  <anyname>*.template  add
#          <anyname>_template          =          <templatename>
include          $(TOP)/configure/RULES
#-----
#          ADD          RULES          AFTER          THIS          LINE
../O.Common/can_ifc.template:          $(DBTEMPLATES)/can_ifc.template.pre
cd ../O.Common && $(DBTEMPLATES)/replaceCmds.pl -I$(CAN_INCLUDE_DIR)/cps_can.h $< $@

can_ifc.template$(DEP):          ../Makefile
@$(RM)          $@
@echo          "$(COMMONDEP_TARGET):          $(DBTEMPLATES)/replaceCmds.pl"          >          $@
@echo          "$(COMMONDEP_TARGET):          $(CAN_INCLUDE_DIR)/cps_can.h"          >>          $@
@echo          "$(COMMONDEP_TARGET):          $(DBTEMPLATES)/can_ifc.template.pre"          >>          $@

../O.Common/qm07card.template:          $(DBTEMPLATES)/qm07/qm07card.template.pre
cd ../O.Common && $(DBTEMPLATES)/replaceCmds.pl -I $(CAN_INCLUDE_DIR)/cps_can.h $< $@

qm07card.template$(DEP):          ../Makefile
@$(RM)          $@
@echo          "$(COMMONDEP_TARGET):          $(DBTEMPLATES)/replaceCmds.pl"          >          $@
@echo          "$(COMMONDEP_TARGET):          $(CAN_INCLUDE_DIR)/cps_can.h"          >>          $@
@echo          "$(COMMONDEP_TARGET):          $(DBTEMPLATES)/qm07/qm07card.template.pre"          >>          $@

../O.Common/qm07_iadc.template:          $(DBTEMPLATES)/qm07/qm07_iadc.template.pre
cd ../O.Common && $(DBTEMPLATES)/replaceCmds.pl -I $(CAN_INCLUDE_DIR)/cps_can.h $< $@

qm07_iadc.template$(DEP):          ../Makefile
@$(RM)          $@
@echo          "$(COMMONDEP_TARGET):          $(DBTEMPLATES)/replaceCmds.pl"          >          $@
@echo          "$(COMMONDEP_TARGET):          $(CAN_INCLUDE_DIR)/cps_can.h"          >>          $@
@echo          "$(COMMONDEP_TARGET):          $(DBTEMPLATES)/qm07/qm07_iadc.template.pre"          >>          $@

../O.Common/qm07_iadc_sum.template:          $(DBTEMPLATES)/qm07/qm07_iadc_sum.template.pre
cd ../O.Common && $(DBTEMPLATES)/replaceCmds.pl -I $(CAN_INCLUDE_DIR)/cps_can.h $< $@

qm07_iadc_sum.template$(DEP):          ../Makefile

```

```

    @$ (RM)                                     @$
    @echo    "$ (COMMONDEP_TARGET):    $(DBTEMPLATES)/replaceCmds.pl"    >    @$
    @echo    "$ (COMMONDEP_TARGET):    $(CAN_INCLUDE_DIR)/cps_can.h"    >>    @$
    @echo    "$ (COMMONDEP_TARGET):    $(DBTEMPLATES)/qm07/qm07_iadc_sum.template.pre"    >>    @$

../O.Common/qm07crate.template:                $(DBTEMPLATES)/qm07/qm07crate.template.pre
    cd ../O.Common && $(DBTEMPLATES)/replaceCmds.pl -I $(CAN_INCLUDE_DIR)/cps_can.h $< @$

qm07crate.template$(DEP):                      ../Makefile
    @$ (RM)                                     @$
    @echo    "$ (COMMONDEP_TARGET):    $(DBTEMPLATES)/replaceCmds.pl"    >    @$
    @echo    "$ (COMMONDEP_TARGET):    $(CAN_INCLUDE_DIR)/cps_can.h"    >>    @$
    @echo    "$ (COMMONDEP_TARGET):    $(DBTEMPLATES)/qm07/qm07crate.template.pre"    >>    @$

../O.Common/qm07_avgChannel.template:          $(DBTEMPLATES)/qm07/qm07_avgChannel.template
    cd ../O.Common && $(DBTEMPLATES)/replaceCmds.pl -I $(CAN_INCLUDE_DIR)/cps_can.h $< @$

qm07_avgChannel.template$(DEP):               ../Makefile
    @$ (RM)                                     @$
    @echo    "$ (COMMONDEP_TARGET):    $(DBTEMPLATES)/replaceCmds.pl"    >    @$
    @echo    "$ (COMMONDEP_TARGET):    $(CAN_INCLUDE_DIR)/cps_can.h"    >>    @$
    @echo    "$ (COMMONDEP_TARGET):    $(DBTEMPLATES)/qm07/qm07_avgChannel.template"    >>    @$

../O.Common/spektrometer_cup_alert.template:  $(DBTEMPLATES)/../Spektrometer/spektrometerApp/Db/
spektrometer_cup_alert.template
    cd ../O.Common && $(DBTEMPLATES)/replaceCmds.pl -I $(CAN_INCLUDE_DIR)/cps_can.h $< @$

limit_cup.template$(DEP):                    ../Makefile
    @$ (RM)                                     @$
    @echo    "$ (COMMONDEP_TARGET):    $(DBTEMPLATES)/replaceCmds.pl"    >    @$
    @echo    "$ (COMMONDEP_TARGET):    $(CAN_INCLUDE_DIR)/cps_can.h"    >>    @$
    @echo    "$ (COMMONDEP_TARGET):    $(DBTEMPLATES)/../Spektrometer/spektrometerApp/Db/
spektrometer_cup_alert.template"    >>    @$

force:

```

- acs/frameworks/epics/iocapps/Spektrometer/trunk/spektrometerApp/src/Makefile

```

TOP=../..

include                                           $(TOP)/configure/CONFIG
#-----
#          ADD      MACRO      DEFINITIONS      AFTER      THIS      LINE
#=====

#-----
#          Build      the      IOC      application

PROD_IOC
#      spektrometer.dbd      will      be      created      and      installed
DBD      +=      spektrometer.dbd

#      spektrometer.dbd      will      be      made      up      from      these      files:
spektrometer_DBD      +=      base.dbd

```

```

spektrometer_DBD                                     +=                                     tudSocketCan.dbd

#   Include   dbd   files   from   all   support   applications:
#spektrometer_DBD                                     +=                                     xxx.dbd

#   Add   all   the   support   libraries   needed   by   this   IOC
spektrometer_LIBS                                     +=                                     tudSocketCan

#   spektrometer_registerRecordDeviceDriver.cpp   derives   from   spektrometer.dbd
spektrometer_SRCS                                     +=                                     spektrometer_registerRecordDeviceDriver.cpp

#   Build   the   main   IOC   entry   point   on   workstation   OSs.
spektrometer_SRCS_DEFAULT                             +=                                     spektrometerMain.cpp
spektrometer_SRCS_vxWorks                             +=                                     -nil-

#   Add   support   from   base/src/vxWorks   if   needed
#spektrometer_OBJS_vxWorks                             +=                                     $(EPICS_BASE_BIN)/vxComLibrary

#   Finally   link   to   the   EPICS   Base   libraries
spektrometer_LIBS                                     +=                                     $(EPICS_BASE_IOC_LIBS)

#=====

include                                             $(TOP)/configure/RULES
#-----
# ADD RULES AFTER THIS LINE

```

- acs/frameworks/epics/iocapps/
Spektrometer/trunk/spektrometerApp/src/spektrometerMain.cpp

```

/*          spektrometerMain.cpp          */
/*   Author:      Marty   Kraimer   Date:      17MAR2000   */

#include <stddef.h>
#include <stdlib.h>
#include <stddef.h>
#include <string.h>
#include <stdio.h>

#include "epicsExit.h"
#include "epicsThread.h"
#include "iocsh.h"

int          main(int          argc,char          *argv[])
{
    if(argc>=2)
        iocsh(argv[1]);
        epicsThreadSleep(.2);
}

```

```

}
iocsh(NULL);
epicsExit(0);
return(0);
}

```

Um die neuen templates verwenden zu können mussten Ersetzungsregeln hinzugefügt werden, diese sind in der Datei spektrometer.substitutions enthalten. Die templates qm07_iadc.template.pre und qm07_iadc_sum.template.pre benötigen die Adressierungen der Messkanäle, welche bis auf die chanID für beide Spektrometer übereinstimmen. Weitere Informationen über die Adressierung können in [18, 22] gefunden werden. In den Ersetzungsregeln für qm07avgChannel sind die Zeit (averaging_time), über die gemittelt wird, und die tolerierte Differenz zwischen zwei gemittelten Werten (ADEL) enthalten. Es ist wichtig, dass der Name in qm07crate.template mit der Bezeichnung für die PU-Einheit in qm07_iadc_sum.template.pre übereinstimmt, da sonst die Berechnung der Ladung ein Fehler auftritt. In den Testläufen ist aufgefallen, dass bei der Bestimmung der Ladung nach mehreren Stunden ein Fehler auftritt. Die Werte für Strom und gemittelten Strom wurden jedoch weiterhin korrekt ausgegeben. Möglicherweise liegt die Ursache darin, dass die PU-Einheit Anfragen schneller bekommt als sie bearbeitet werden können [23]. Um dies zu verhindern wurde entschieden die Anzahl von zehn Anfragen pro Sekunde nach Zeit und Messzyklen auf eine pro Sekunde zu reduzieren. Der Abstand zwischen zwei Anfragen war in der Datei qm07crat.template.pre fest vorgegeben. Da dieses Template auch von anderen IOC verwendet wird, wurde der vorgegebene Wert durch den Platzhalter scan_time_cycles ersetzt. Diesem Platzhalter muss in spektrometer.substitutions ein Wert zugewiesen werden. Auch für das Template spektrometer_cup_alert.template müssen Ersetzungsregeln vorgegeben werden.

- acs/frameworks/epics/iocapps/spektrometer/trunk/spektrometerApp/Db/spektrometer.substitutions

```

file "db/can_ifc.template" {
  {
    name = "Spektrometer:Cup:Ifc"
    can_interface = "can1"
    crate_id = "02"
    slot_id = "1"
    scan_fast_single_device = "1 second"
    scan_slow_single_device = "10 second"
  }
}

file "db/qm07crate.template" {
  {
    name = "Spektrometer:Cup"
    can_interface = "can1"
    crate_id = "02"
    scan_fast_single_device = "1 second"
    scan_slow_single_device = "10 second"
    scan_time_cycles = ".2 second"
  }
}

```

```

}

file "db/qm07card.template"
{
    pattern
    { name          card_id          card  can_interface crate_id
    { Spektrometer  slot01          00   can1          2
}

file "db/qm07_iadc.template"
{
    pattern
    { name          can_interface crate_id  chanID}
    { Lintott:Cup  can1          02     0x00  }
    { QClam:Cup   can1          02     0x01  }
}

file "db/qm07_iadc_sum.template"
{
    pattern
    { name          can_interface crate_id  chanID  pu
    { Lintott:Cup  can1          02     0x00   Spektrometer:Cup  }
    { QClam:Cup   can1          02     0x01   Spektrometer:Cup  }
}

file "db/qm07_avgChannel.template"
{
    pattern
    { name          srcRecord          averaging_time  ADEL  enabledByDefault }
    { Lintott:Cup  Lintott:Cup:rdIADC  "1 second"     0.5   1
    { QClam:Cup   QClam:Cup:rdIADC  "1 second"     0.5   1
}

file „db/spektrometer_cup_alert.template“
{
    { name          }
    { Lintott:Cup  }
    { QClam:Cup   }
}

```

- `acs/frameworks/epics/iocapps/templates/trunk/qm07/qm07crate.template.pre`
(Auszug mit den Änderungen)

```

record(longout, "$(name):reqTime") {
    field(SCAN, "$(scan_time_cycles)")
    field(DTYP, "tudSocketCan")
    field(OUT, "@$(can_interface) 06 $(crate_id) 2 QM07_GET_DATA 0xFC 1 u0 5")
}

record(longout, "$(name):reqCycles") {
    field(SCAN, "$(scan_time_cycles)")
    field(DTYP, "tudSocketCan")
}

```

```
field(OUT, "@$(can_interface) 06 $(crate_id) 2 QM07_GET_DATA 0xFD 1 u0 5")
}
```

Mit dem Programm CSS (Control System Studio) können operator interfaces (OPIs) erstellt werden mit welchen es möglich ist dem IOC Daten zu übergeben oder Daten des IOC auszulesen.

Nachdem der benötigte Treiber installiert ist [22] kann der IOC kompiliert und gestartet werden. Zum kompilieren muss im Ordner `acs/frameworks/epics/iocapps/<IOC NAME>/trunk` der Befehl `make` ausgeführt werden. Nun kann der IOC mit der Datei `st.cmd` im Ordner `acs/frameworks/epics/iocapps/<IOC NAME>/trunk/iocBoot/ioclinux-x86`(für 64 Bit Systeme `_64` anhängen) gestartet werden.

Tabelle 4: Aufgelistet sind die wichtigen Prozess Variablen und deren Verwendungszweck.

Prozess Variable	Verwendung
Lintott:Cup:rdIADC	Strom am Lintott Faraday-Cup
QClam:Cup:rdIADC	Strom am QClam Faraday-Cup
Lintott:Cup:avgValue	Gemittelter Strom am Lintott Faraday-Cup
QClam:Cup:avgValue	Gemittelter Strom am QClam Faraday-Cup
Lintott:Cup:rdIADCsum	Ladung am Lintott Faraday-Cup
QClam:Cup:rdIADCsum	Ladung am QClam Faraday-Cup
Lintott:Cup:alert	Grenzwert für Strom am Lintott Faraday-Cup
QClam:Cup:alert	Grenzwert für Strom am QClam Faraday-Cup

5.7. Datenaufnahme des Strahlstroms mit MBS

Die benötigten Dateien für die Stromauslese mit MBS wurden aus den Projekten „(e,e’x) last“ und Silizium Ball übernommen. Von dem Projekt „(e,e’x) last“ wurden die Dateien `f_user.c` sowie zwei Header für die Adressierung (`create_mem.h`) und Auslese des Scalers (`sis_3820.h`). Die restlichen erforderlichen Dateien (`setup.usf`, `startup.scom` und `shutdown.scom`) wurden von dem Projekt Silizium Ball übernommen. Die Skripte `startup.scom` und `shutdown.scom` werden zu Beginn bzw. am Ende einer Messung ausgeführt, `setup.usf` wird von diesen Skripten benötigt. An den Dateien aus dem Projekt „(e,e’x) last“ mussten noch Anpassungen vorgenommen werden, dies geschah in drei Schritten. Im ersten Schritt wurden alle Module und nicht benötigte Verweise außer dem Scaler aus der Datei `f_user.c` entfernt. Der zweite Schritt bestand darin die Initialisierung und Auslese des Scalers anzupassen. Dies war notwendig, da der Scaler für eine Zeitmessung im Preset Modus betrieben wurde. Anschließend wurde im dritten Schritt der Code übersichtlicher gestaltet. Dafür wurde eine Datei `sis_3820.c` erstellt, welche Funktionen für die Initialisierung und Auslese des Scalers beinhaltet. Als Vorlage dafür diente das Projekt Silizium Ball [26].

5.7.1. f_user.c

Die Datei `f_user.c` beinhaltet drei Funktionen [26]:

- `f_user_get_virt_ptr`: Setzt virtuelle Pointer zu Adressen der Module
- `f_user_init`: Initialisiert die Module
- `f_user_readout`: Verarbeitet Trigger Signale und liest Module aus.

In der Funktion `f_user_get_virt_ptr` wird die `create_mem()` Funktion des Headers `smem.h` ausgeführt, dies ermöglicht die Adressierung des Scalers. Anschließend werden die virtuellen Pointer des Scalers erzeugt. Die Funktion `f_user_init` führt die Funktion `sis_3820_init` aus, dadurch wird der Scaler initialisiert. Das Auslesen des Scalers geschieht in der Funktion `sis_3820_read`, welche in der Funktion `f_user_readout`, bei dem Triggersignal 1 ausgeführt wird.

```

/*****/
/* Setup for proton radius experiment */
/* - specific for CES RIO3 8064 PowerPC-Based VME board */
/*****/
/* Authors: */
/* N. Kurz 10.11.1995 (template) GSI Darmstadt */
/* A. Wagner 17.01.2005 FZ Rossendorf */
/* R. Beyer 18.05.2005 FZ Rossendorf */
/* I. Pysmenetska 30.05.2006 Darmstadt IKP */
/* G. Steinhilber 06.11.2013 Darmstadt IKP */
/*****/

// 20060717 Art adding setting thresholds from a parameter file
// 20060718 Art adding output of thresholds for Case 14- start acq
// Also adding output of statistics for Case 15- stop acq
// 20131106 Removed all Modules except SIS3820 Scaler and changed how the Scaler works

/*****/
/* Parameters specific to the crate stuffing */
/*****/

#define BASE_SIS 0x09000000 // base of SIS 3820 Scaler

/*****/
#include <stdio.h>
#include <s_yeshe.h>
#include <smem.h>

#include "create_mem.h" // memory segment for static memory mapping

#include "modules/sis_3820.h" // Scaler SiS 3820

//static struct
static struct SIS3820 *p_sis; // pointer to SiS 3820 Scaler

/*****/
/* all pointers which are used for read/write operations in the functions */

```

```

/* f_user_init and f_user_readout must be defined here as static variables */
/*****/

/*****/
/* create virtual pointer to be used in f_user_init and f_user_readout */
/*****/

int f_user_get_virt_ptr (long *pl_loc_hwacc, long pl_rem_cam[])
{
    create_mem(); // make memory segments for static memory mapping

    printm("vital pointers SIS scaler");

    //----- SIS SCALER -----
    p_sis = (SIS3820 *) ( (unsigned int) pl_a32 + (unsigned int)BASE_SIS);
} // created virtual pointers

/*****/
/* f_user_init is executed once after start acquisition */
/*****/

int f_user_init (unsigned char bh_crate_nr,
                long *pl_loc_hwacc,
                long *pl_rem_cam,
                long *pl_stat)
{
    switch (bh_crate_nr)
    {
        case 0:
        {
            sis_3820_init(p_sis);
        }
        break;
        default:
        break;
    } //switch case 0

    printm("\nInitialization done!!!\n");
    fflush(stdout);
    return (1);
} // end of initialization

```

```

/*****
/* READOUT */
*****/

int f_user_readout (unsigned char bh_trig_typ,
                   unsigned char bh_crate_nr,
                   register long *pl_loc_hwacc, /* pointer on vme area */
                   register long *pl_rem_cam, /* pointer on camac area */
                   long *pl_dat, /* pointer on data area */
                   s_veshe *ps_veshe, /* pointer on header */
                   long *l_se_read_len, /* number of data bytes */
                   long *l_read_stat)
{
    *l_se_read_len = 0;

    switch (bh_crate_nr)
    {
        case 0:
        {
            switch (bh_trig_typ)
            {
                case 1:
                {
                    //readout first channel of SIS 3820 Scaler
                    *l_se_read_len += sis_3820_read(pl_dat, 1, p_sis);
                    break;
                }
                case 2:
                case 3:
                case 4:
                case 5:
                case 6:
                case 7:
                case 8:
                case 9:
                case 10:
                case 11:
                case 12:
                case 13:
                case 14: //sta acq
                case 15: //sto acq
                default:
                    break;
            }
        }
        break;
    }
}

```

```

        default:
        break;
    }

    return 1;
} // end of readout

```

5.7.2. sis_3820.h

Der Header sis_3820.h enthält eine Struktur für den Scaler, diese Struktur wird in der Datei f_user.c initialisiert. Zudem werden die, in sis_3820.c enthaltenen, Funktionen aufgelistet.

```

// 20060607 A. Shevchenko
// 20131106 G. Steinhilber

// control structure for SIS 3820 SCALER

#ifdef SIS_3820_H
#define SIS_3820_H

#include <stdio.h>
#include <string.h>

// bits for Control Register cstat
#define SIS_ULED 0x01 ; // bit 0 - User LED, 1:on
#define SIS_25MHZ 0x10 ; // bit 4 - 25MHz Test pulses
#define SIS_TEST 0x20 ; // bit 5 - Test mode
#define SIS_REF 0x40 ; // bit 6 - Ch1 as Reference internal 50MHz

// bits of preset_en Register
#define SIS_GR1_EN 0x01 ; // bit 0 Enable Group 1
#define SIS_GR2_EN 0x10000 ; // bit 16 Enable Group 2

// bits for OP_MODE Register
#define SIS_NON_CLEAR 0x01; // bit 0 NON-Clearing mode
#define SIS_OPMODE_0 0x00; // bits 28-30 000 Scaler/Counter
#define SIS_OPMODE_1 0x20000000 // bits 28-30 010 Multichannel scaler
#define SIS_OPMODE_2 0x70000000 // bits 28-30 111 SDRAM VME write test
#define SIS_LNE_SRC_PRESET 0x40; // bits 4-6 100 Preset Scaler N ->LNE Source
#define SIS_IMODE_0 0x0 // bits 16-18 000 all inputs no function
#define SIS_IMODE_1 0x10000 // 001 Mode1: 1-ext next,2-user1,3-user2,4-inhLNE
#define SIS_IMODE_2 0x20000 // 010 Mode2: 1-extnext,2-user1,3-inhCNT,4-inhLNE
#define SIS_IMODE_3 0x30000 // 011 Mode3: 1-extnext,2-user1,3-user2,4-inhCNT
#define SIS_IMODE_4 0x40000 // 100 Mode4: 1-inh(0-7),2-inh,3-inh3,4-inh(24-31)
#define SIS_IMODE_INV 0x80000 //b19 Inputs 1-4 Inverted?
#define SIS_OMODE_0 0x0 //b20-21 00 M0: 5-LNE,6-SDRAMempty,7-SDRAMthr,8-userLED
#define SIS_OMODE_1 0x100000 //b20-21 01: M1: 5-LNE,6-Enable,7-50MHz,8-UserLED
#define SIS_OMODE_INV 0x800000 //b23 OUtputs 5-8 Inverted?

```

```

#ifdef u_long
#undef u_long
#endif

#ifndef u_long
#define u_long unsigned long
#endif

struct SIS3820
{
    volatile u_long    cstat;        //0x0000 rw: Control/Status Register
    volatile u_long    firmware;     //0x0004 R:  firmware version
    volatile u_long    ir_config;    //0x0008 rw: Interrupt Configuration
    volatile u_long    ir_cstat;     //0x000C rw: Interrupt Control/Status
    volatile u_long    acq_preset;   //0x0010 rw: ACQ Preset Register
    volatile u_long    acq_cnt;      //0x0014 R:  ACQ Count Register
    volatile u_long    lne_prescale; //0x0018 rw: LNE Prescale factor

    volatile u_long    dumm1;        //0x001C -

    volatile u_long    preset1;      //0x0020 rw: Preset value register counter group
1
    volatile u_long    preset2;      //0x0024 rw: Preset value register counter group
2
    volatile u_long    preset_en;    //0x0028 rw: Preset enable and hit register

    volatile u_long    dumm2;        //0x002C -

    volatile u_long    cblt;         //0x0030 rw: CBLT/Broadcast setup register
    volatile u_long    sdram_page;   //0x0034 rw: SDRAM page register
    volatile u_long    fifo_cnt;     //0x0038 rw: FIFO Word count register
    volatile u_long    fifo_thr;     //0x003C rw: FIFO Word count Threshold
register

    volatile u_long    dumm3[48];    //0x0040-0x0100 -

    volatile u_long    op_mode;      //0x0100 rw: (ACQ) Operation mode register
    volatile u_long    copy_dis;     //0x0104 rw: copy disable register
    volatile u_long    lne_ch;       //0x0108 rw: LNE channel select (1 of 32)
    volatile u_long    preset_ch;    //0x010C rw: PRESET channel select (2 time 1
out of 16)

    volatile u_long    dumm4[60];    //0x0110-0200 -

    volatile u_long    inh;          //0x0200 rw: Inhibit/count disable register
    volatile u_long    cnt_clr;      //0x0204 W:  Counter Clear register
    volatile u_long    overflow;     //0x0208 rw: Counter Overflow read/Clear

```

```

    volatile u_long    johnson;    //0x020C R: Johnson error register (SIS
internal)

    volatile u_long    dumm5[124]; //0x0210-0400 -

    volatile u_long    key_rst;    //0x0400 ??: Key Reset
    volatile u_long    key_fifo_rst; //0x0404 ??: Key SDRAM/FIFO Reset
    volatile u_long    key_test;    //0x0408 ??: Key Test pulse
    volatile u_long    key_clr;    //0x040C ?????: Key Counter Clear OR
0x0414??????
    volatile u_long    key_shadow; //0x0410 ??: Key VME LNE/clock shadow
    volatile u_long    op_arm;     //0x0414 ??: Key operation arm
    volatile u_long    op_en;     //0x0418 ??: Key operation enable
    volatile u_long    op_dis;    //0x041C ??: Key operation disable

    volatile u_long    dumm6[248]; //0x0420-0800 rw: -

    volatile u_long    shadow[32]; //0x0800- R: Shadow Registers

    volatile u_long    dumm7[96]; //0x0880-A00 : -

    volatile u_long    cnt[32];    //0x0A00 R: Counter registers ; FIFO address
space in emulat

    //volatile u_long sdram;    //0x800000 rw: SDRAM or FIFO space array. One
8Mbyte page

};    // length of the structure must be ?

typedef struct SIS3820 SIS3820;

/*
 * Functions in sis_3820.c
 */

//Initializes SIS 3820 Scaler
void sis_3820_init( SIS3820 *p_sis);

//Read SIS 3820 Scaler
int sis_3820_read(long *pl_dat, int number_channel, SIS3820 *p_sis);

#endif

```

5.7.3. sis_3820.c

Die Datei sis_3820 beinhaltet zwei Funktionen:

- sis_3820_init: Initialisiert den Scaler
- sis_3820_read: Liest den Scaler aus

Die Funktion `sis_3820_init` setzt alle Kanäle des Scalers auf null zurück, legt den Betriebsmodus fest und gibt die Kanäle frei, damit einlaufende Rechtecksignale gezählt werden können. Das Auslesen des Scalers erfolgt in der Funktion `sis_3820_read`, die Länge des zu belegenden Speichers wird dabei mit der `return` Anweisung zurückgegeben.

```
#include "sis_3820.h"

// Initialization of SIS 3820 Scaler
void sis_3820_init(SIS3820 *p_sis)
{
    int l_j = 0;

    printm("Init SIS Scaler\n");
    fflush(stdout);
    printf("firmware:    %x\n", p_sis->firmware );
    fflush(stdout);

    p_sis->op_dis = 0x0; // disable counting
    p_sis->inh = 0x0; // activate all channels
    p_sis->key_rst = 0x0; // Key Reset

    p_sis->cnt_clr = 0xFFFFFFFF; // clear all counters
    p_sis->op_mode = 0x01 + SIS_OMODE_0;

    p_sis->op_arm = 0x00; //scaler -----

    for (l_j=0; l_j<32; l_j++)
    {
        printm("SIS ch %d:    cnt: 0x%x  shadow:%lx \n", l_j, p_sis->cnt[l_j], p_sis-
>shadow[l_j]);
    }

    (*p_sis).op_en = 0x0; // enable counting SIS Scaler
}

// Readout SIS 3820 Scaler
int sis_3820_read(long *pl_dat, int number_channel, SIS3820 *p_sis)
{
    int l_j = 0;
    long *pl_dat_save = pl_dat;

    p_sis->op_dis = 0x0; // disable counting
    *pl_dat++ = 0x40000000 + (32+1); //creating header wordoe

    for (l_j=0; l_j<number_channel; l_j++) //for each channel
    {
        *pl_dat++ = p_sis->cnt[l_j]; //read counter
    }
}
```

```

        printf("SIS ch %d: cnt: 0x%x shadow:%lx \n", l_j, p_sis->cnt[l_j], p_sis-
>shadow[l_j]);
    }

    *pl_dat++ = 0xC0000000; //creating end-of-event word
    p_sis->op_en = 0x0; // enable counting SIS Scaler

    return (char*)pl_dat - (char*)pl_dat_save;
}

```

5.7.4. Makefile

Das Makefile wurde aus dem Projekt Silizium Ball übernommen, angepasst werden musste lediglich der Verweis auf das verwendete Modul.

```

# N.Kurz, GSI, 16-Aug-2001

vpath %.c modules
vpath %.h modules

%.o: %.c %.h
    gcc -c $<

MODULES = sis_3820.o

INC = $(MBSROOT)/inc
OBJ = $(MBSROOT)/obj_$(GSI_CPU_PLATFORM)
LIB = $(MBSROOT)/lib_$(GSI_CPU_PLATFORM)

SMILIBS = $(LIB)/lib_mbs.a $(LIB)/lib_tools.a $(LIB)/lib_mbs.a $(LIB)/lib_smi.a
CES_RIO2_LIB = -L/lib/ces -lvme -lbma -luio

SHMQ = -DLYNX__SMEM
ifeq ($(GSI_OS_VERSION),2.5)
POSIX = -mthreads
CFLAGS = $(POSIX) -Dlynx -DGSI_MBS -D$(GSI_CPU_ENDIAN) -D$(GSI_CPU_PLATFORM)
LIBS = $(LIB)/lib_mbs.a $(LIB)/lib_tools.a $(LIB)/lib_mbs.a \
        $(LIB)/libedit.a -llynx -lnetinet
endif

ifeq ($(GSI_OS_VERSION),3.0)
POSIX = -mthreads
CFLAGS = $(POSIX) -D_THREADS_POSIX4ad4 -Dlynx -DGSI_MBS -D$(GSI_CPU_ENDIAN)
-D$(GSI_CPU_PLATFORM)
LIBS = $(LIB)/lib_mbs.a $(LIB)/lib_tools.a $(LIB)/lib_mbs.a \
        $(LIB)/libedit.a -llynx -lnetinet
endif

```

```
ifeq ($(GSI_OS_VERSION),3.1)
POSIX = -mthreads
CFLAGS = $(POSIX) -D_THREADS_POSIX4ad4 -DLynx -DGSI_MBS -D$(GSI_CPU_ENDIAN)
-D$(GSI_CPU_PLATFORM)
LIBS = $(LIB)/lib_mbs.a $(LIB)/lib_tools.a $(LIB)/lib_mbs.a \
      $(LIB)/libedit.a -llynx -lposix-pre1c -lnetinet
endif

m_read_meb : $(OBJ)/m_read_meb.o $(OBJ)/objects f_user.c $(MODULES)
      gcc -O2 $(CFLAGS) -o m_read_meb -I$(INC) f_user.c $(MODULES)
$(OBJ)/m_read_meb.o $(LIBS)
      strip m_read_meb

clean :
      rm -f core *.bak *.jou m_read_meb mbslog.l $(MODULES)
```

Literaturverzeichnis

- [1] A. Richter, „Operational experience at the S-DALINAC“, Darmstadt, 1996.
- [2] A. Krugmann, „Untersuchung von E0-Übergängen in Kernen am Phasenübergang zwischen sphärischer und deformierter Kerngestalt“, Darmstadt, 2008.
- [3] F. Neumeyer, „Untersuchung magnetischer Kernanregungen in ^{48}Ca und ^{90}Zr mit hochauflösender Elektronenstreuung unter 180° am S-DALINAC“, Darmstadt, 1997.
- [4] B. Reitz, „Weiterentwicklung des Detektorsystems am QCLAM-Spektrometer des S-DALINAC und Untersuchung der Reaktion $^{48}\text{Ca}(e,e')$ und $^{58}\text{Ni}(e,e')$ unter 180° “, Darmstadt, 2000.
- [5] G. Herbert, „Entwurf und Aufbau des Meßdatenerfassungssystems am supraleitenden Darmstädter Elektronenbeschleuniger S-DALINAC und statistische Analyse von $^{40}\text{Ca}(e,e'x)$ -Experimenten“, Darmstadt, 1994.
- [6] A. Byelikov, „Neutrino-Nukleosynthese der seltenen Isotope ^{138}La und ^{180}Ta und Entwicklung eines Siliziumballs für exklusive Elektronenstreuexperimente am S-DALINAC“, Darmstadt, 2007.
- [7] Y. Poltoratska, C. Eckardt, W. Ackermann, K. Aulenbacher, T. Bahlo, R. Barday, M. Brunken, C. Burandt, R. Eichhorn, J. Enders, M. Espig, S. Franke, C. Ingenhaag, J. Lindemann, W.F.O. Müller, M. Platz, M. Roth, F. Schneider, M. Wagner, A. Weber, T. Weiland, B. Zwicker, "developments at the polarized-electron injector of the superconducting Darmstadt electron linear accelerator", *Journal of Physics: Conference Series*, 2011.
- [8] C. Eckardt, R. Barday, U. Bonnes, M. Brunken, R. Eichhorn, J. Enders, C. Heßler, W. F. O.Müller, M. Platz, Y. Poltoratska, B.Steiner, M. Wagner, and T. Weiland, „Polarimetry at the Superconducting Darmstadt Electron Linac S-DALINAC,“ in *AIP Conference Proceedings*, 2009, pp. 1149-1170.
- [9] M. Knirsch, „Konzeption, Aufbau und erprobung eines hochauflösenden QCLAM-Elektronenspektrometers mit großem Raumwinkel und hoher Impulsakzeptanz am Elektronenbeschleuniger S-DALINAC“, Darmstadt, 1991.
- [10] C. Lüttge, „Entwicklung und Aufbau eines Magnetsystems für Elektronenstreuung unter 180° und vollständige Bestimmung der magnetischen Dipol- und Quadrupolstärke-verteilung in ^{28}Si “, Darmstadt, 1994.
- [11] K.-D. Hummel, „Entwicklung und Inbetriebnahme eines Vieldrahtdriftkammer-Detektorsystems für das QCLAM-Spektrometer am supraleitenden Darmstädter Elektronenbeschleuniger S-DALINAC“, Darmstadt, 1992.
- [12] R. Kämpf, „Das Trigger-Detektorsystem des neuen QCLAM Spektrometers am S-DALINAC und Entwicklung eines CAMAC-Moduls zur optischen Datenübertragung“, Darmstadt, 1991.
- [13] S. Aslanidou, *Quelle unbekannt*.
- [14] M. Hess, 2013, Persönliche Mitteilung.
- [15] SPS Lehrgang, Aufruf: 28.9.2012, <http://www.sps-lehrgang.de/geschichte-der-sps/>.
- [16] R. Smazinka, Aufruf: 28.9.2012 http://www.smazinka.de/wasist_sps.html.
- [17] A. Araz, „Aufbau und Erprobung einer digitalen HF-Regelung und Aufbau eines modularen Messsystems zur Energiestabilisierung für den S-DALINAC.“, TU Darmstadt,

-
- Darmstadt, 2009.
- [18] S. Bassauer, „Weiterentwicklung der Online-Datenanalyse sowie der Ladungs- und Stromauslese am LINTOTT-Spektrometer“, Darmstadt, 2012.
- [19] Uwe Bonnes, 2013, Persönliche Mitteilung.
- [20] Experimental Physics and Industrial Control System, Aufruf: 11.2013: „<http://aps.anl.gov/epics>“.
- [21] Control System Studio, Aufruf: 30.11.2013, <http://controlsystemstudio.github.io/>.
- [22] H. Lauinger, „Controlling the Goniometer at QCLAM with EPICS“, Darmstadt, 2012.
- [23] C. Burandt, 10.2013, Persönliche Mitteilung.
- [24] T. Bahlo, 11.2013, Persönliche Mitteilung.
- [25] Go4, Aufruf: 11.11.2013, <http://web-docs.gsi.de/~go4/download/go4.php>.
- [26] H. Lauinger, „Aufbau und Inbetriebnahme des Silizium Detektor-Balls am QCLAM Spektrometer“, Darmstadt, 2012.
- [27] Beschleunigerwiki, Aufruf: 12.2013, ikpweb.ikp.tu-darmstadt.de.
- [28] SIS GmbH, „SIS3820 VME Scaler User Manual“, Hamburg, 2007.
- [29] J. Birkhan, 10.2013, Persönliche Mitteilung.
- [30] S. Bassauer, 10.2013, Persönliche Mitteilung.
- [31] EATON, Aufruf: 30.11.2013, http://www.moeller.net/de/products_solutions/motor_applications/control/easy/software.jsp.

Danksagung

Bedanken möchte ich mich bei den Professoren Pietralla und von Neuman-Cosel für die Möglichkeit mich mit dem S-DALINAC sowie den Spektrometern zu beschäftigen.

Meiner Betreuerin Dipl.-Phys. Simela Aslanidou, sowie Dipl.-Phys. Jonny Birkhan und Andreas Köhler, M. Sc., welche mich tatkräftig unterstützt haben, danke ich ebenfalls.

Weiterhin möchte ich mich bei den Leitern der Mechanik-Werkstatt, Herrn Oppermann, und der Elektronik Werkstatt, Dipl.-Ing. Uwe Bonnes, bedanken. Beide haben mir in Gesprächen viel Wissen vermittelt. Gleiches gilt für Christoph Burandt, M. Sc., sowie Manfred Hess und die gesamte Betriebsgruppe.

Des weiteren möchte ich mich bei Mirko von Schmitt, M. Sc., Maxim Singer, B. Sc., und Dipl.-Phys. Michael Reese bedanken, sie konnten mir viel Wissen über MBS und Go4 vermitteln. Auch bei Sergej Bassauer, B. Sc., bedanke ich mich, er war eine große Hilfe bei der QM07-Elektronik.

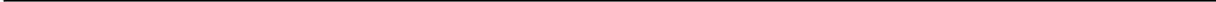
Andreas Krugmann möchte ich für die Gelegenheit viel über das Lintott-Spektrometer danken, ihm bin ich gerne hinterhergerannt, wenn es was am Lintott-Spektrometer zu tun gab.

Ebenfalls bedanke ich mich bei meinen Bürokollegen Alexander Hufnagel und Michael Mathy.

Bedanken möchte ich mich bei meiner Freundin, Michaela Hilcker, sie stand mir immer tröstend zur Seite, wenn mal wieder etwas gar nicht funktioniert hat. Auch den, noch nicht namentlich genannten, Kollegen aus der Spektrometergruppe danke ich für das angenehme Arbeitsklima.

Meiner Familie danke ich für die Unterstützung während des Studiums.

Abschließend möchte ich mich natürlich bei den Katzen bedanken. Und danke für den vielen Fisch.



Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen

Darmstadt, den

(Gerhart Steinhilber)