

# Masterarbeit

## Erste Erfahrung mit der Messelektronik für das P2-Experiment an MESA

von

**Thomas Jennewein**



Masterarbeit in Physik  
vorgelegt dem Fachbereich Physik, Mathematik und Informatik (FB 08)  
der Johannes Gutenberg-Universität Mainz  
am 10.9.2015

1. Gutachter: Prof. Dr. Frank Maas
2. Gutachter: Prof. Dr. Hans-Jürgen Arends



Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Mainz, den 10.9.2015

Thomas Jennwein  
Institut für Kernphysik  
Johannes Gutenberg-Universität Mainz  
Johann-Joachim-Becher-Weg 45  
D-55128 Mainz  
jennwet@students.uni-mainz.de



## Inhaltsverzeichnis

1. Einleitung	1
2. Das P2-Experiment	2
2.1. Schwacher Mischungswinkel	2
2.2. Experimentelles Konzept zu Messung von $\sin^2\Theta_w$	3
3. Photomultiplier Röhren und Spannungsteiler	6
3.1. Allgemeines	6
3.2. Photoeffekt	6
3.3. Verstärkung	8
3.4. Rauschen und Dunkelpulse	9
3.5. Spannungsteiler	10
3.5.1. Spannungsteiler für das P2-Experiment	14
4. Analog Digital Wandler	17
4.1. Funktionsweise	17
4.1.1. Abtastung des Signals	18
4.1.2. Quantisierung	18
4.1.3. Quantisierungskennlinie	18
4.2. Abweichungen zur idealen Umwandlung	19
4.2.1. Nullpunktfehler	19
4.2.2. Verstärkungsfehler	19
4.2.3. Nichtlineare Fehler	19
4.2.4. Stufenfehler	20
4.3. Realisierung	20
4.3.1. Komparator-Verfahren	20
4.3.2. Rückgekoppelte Umsetzer	20
4.3.3. Parallel Umsetzer	21
4.4. Wichtige Kenngrößen	21
4.5. Verwendete Geräte	22
4.5.1. ADC CAEN V965	22
4.5.2. VME QWEAK Modul von TRIUMF	23
5. Messung und Auswertung	25
5.1. Verstärkungsänderung	25
5.2. Rauschverhalten des TRIUMF ADCs	35
5.3. Linearität des TRIUMF ADCs	39

6. Strahlzeit	44
6.1. Strahlzeit 13.3.2015 . . . . .	44
6.2. Strahlzeit 27.5.2015 . . . . .	52
7. Fazit und Ausblick	60
A. Anhang	63
A.1. Programmcode . . . . .	63
B. Abbildungs- und Tabellenverzeichnis	79
C. Quellen	80
D. Danksagung	82

### 1. Einleitung

Im Rahmen der Exzellenz-Initiative des Bundes wurde an der Johannes Gutenberg-Universität das Exzellenz-Cluster PRISMA<sup>1</sup> etabliert. Im Zuge dieser Förderung wird in Mainz ein neuer Elektronen-Beschleuniger gebaut. Der Name des neuen Beschleunigers ist MESA, was für Mainzer Energy recovering Superconducting Accelerator steht.

Wie der Name schon sagt, handelt es sich um einen supraleitenden Beschleuniger, welcher für hohe Strahlströme bei niedrigen Energien gebaut wird. Er kann in zwei verschiedenen Betriebszuständen betrieben werden, einmal im Extraced-Beam(EB) Modus und im Energy-Recovering-Linac(ERL) Modus. Der Energy-Recovering-Linac erlaubt die Rückführung seiner Energie nach dem Stoß im Target zu den Beschleunigungskavitäten. Am Target hat der Strahl dann eine Energie von 105MeV bei einem Strahlstrom bis zu 10mA. Der EB Modus ist der klassische Modus mit einem externen Target, nachdem der Strahl nach der Wechselwirkung in einem Beam-Dump gestoppt wird. In diesem Modus kann eine Energie von 155 MeV erzielt werden und einem Strahlstrom von 150 $\mu$ A.

Eines der daran geplanten Experimente ist das P2-Experiment. An diesem soll im EB-Modus der elektroschwache Mischwinkel hochpräzise gemessen werden. Um zu gewährleisten, dass entsprechend kleine Unsicherheiten bei der Messung entstehen, müssen alle Bausteine des Experiments wohlbekannt sein. Im Zuge dieser Arbeit wurde sich deshalb intensiv mit der Ausleseelektronik des Experiments beschäftigt.

In Kapitel 2 findet sich eine Einführung in das P2-Experiment. In Kapitel 3 ist eine Einleitung über die Funktion von Photomultiplier-Röhren und deren Betriebsbedingungen. Kapitel 4 handelt von der Funktionsweise von Analog-Digital-Wandlern und welche dabei in dieser Arbeit verwendet wurden. Kapitel 5 beschäftigt sich mit den eigentlichen Tests und Messungen, die mit der Elektronik im Zuge dieser Arbeit durchgeführt wurden. Zusätzlich gab es während dieser Arbeit zwei Mal die Möglichkeit die Elektronik, während einer Teststrahlzeit, zu testen. Dies ist in Kapitel 6 beschrieben. Im abschließenden Kapitel 7 gibt es eine Zusammenfassung und einen Ausblick auf die aus dieser Arbeit entstandene weiterführende Fragestellung.

---

<sup>1</sup>Precision Physics, Fundamental Interactions and Structure of Matter, befasst sich mit dem Aufbau von Materie und wie diese unser Universum beeinflusst.

## 2. Das P2-Experiment

### 2.1. Schwacher Mischungswinkel

Durch die Vereinheitlichung der Quantenelektrodynamik und der schwachen Wechselwirkung zur sogenannten elektroschwachen Wechselwirkung kann man zeigen, dass  $Z^0$  und  $\gamma$  Bosonen analog zur starken Wechselwirkung Mischzustände von zwei nicht beobachtbaren Ausgangszuständen sind. Dass dies Sinn macht, sieht man durch das Überprüfen der Zerfallskanäle von  $Z^0$ . Diese Mischzustände kann man in einer einfachen Matrix schreiben:

$$\begin{pmatrix} \gamma \\ Z^0 \end{pmatrix} = \begin{pmatrix} \cos\Theta_w & \sin\Theta_w \\ -\sin\Theta_w & \cos\Theta_w \end{pmatrix} \begin{pmatrix} B^0 \\ W^0 \end{pmatrix} \quad (1)$$

Wie man sieht, handelt es sich hierbei um eine Drehmatrix. Der Winkel, um den gedreht wurde, ist der elektroschwache Mischungswinkel, auch Weinbergwinkel genannt.

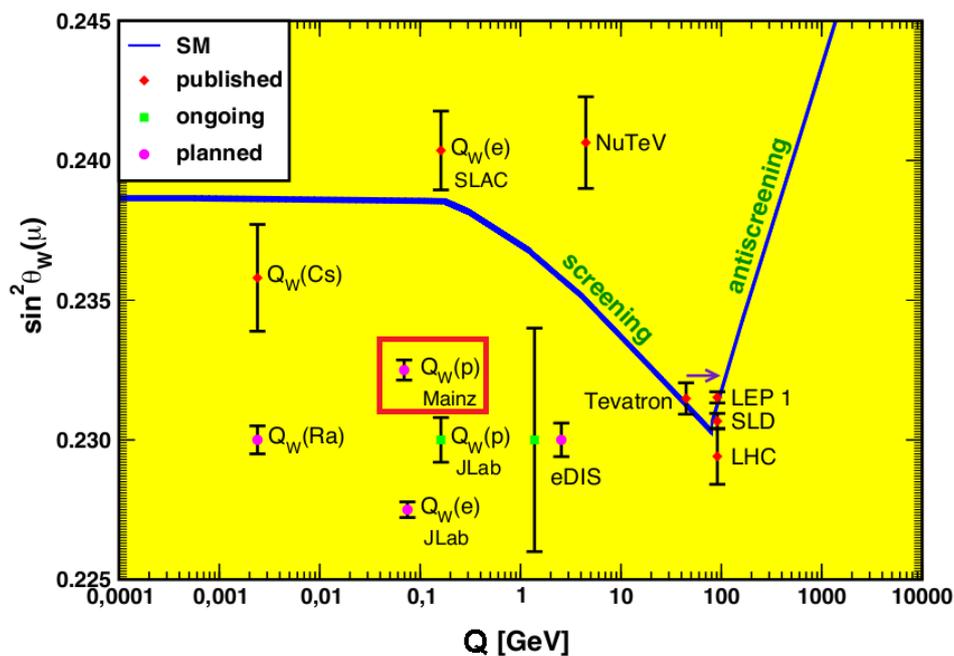


Abbildung freundlicherweise zur Verfügung gestellt von Jens Erler <sup>2</sup>

Abbildung 1:  $\sin^2\Theta_w$  in Abhängigkeit des Impulsübertrags  $Q$ . Universelle Quantenkorrekturen werden in den Mischungswinkel absorbiert, was zu dem in rot gekennzeichneten theoretischen Verlauf aus dem Standardmodell führt. Eingetragen sind bestehende und zukünftige Messungen.

<sup>2</sup>Departamento de Física Teórica, Instituto de Física, Universidad Nacional Autónoma de México

## 2. Das P2-Experiment

---

Im oben stehenden Bild sieht man in der Linie den aus dem Standardmodell vorausgesagten Verlauf des Weinbergwinkels in Abhängigkeit des Impulsübertrags. Das P2 Experiment ist mit einem roten Kasten markiert. Bei allen Messungen mit grünem oder pinkem Datenpunkt ist zu beachten, dass diese noch nicht gemessen wurde, weshalb nur ihre Lage auf der X-Achse relevant ist. In Y-Richtung kann man lediglich den Fehlerbalken im Vergleich zu anderen Experimenten abschätzen. Abweichungen von der theoretischen Vorhersage sind Hinweise dafür, dass das Standardmodell bisher noch nicht komplett ist und somit an dieser Stelle neue Physik gefunden werden kann.

### 2.2. Experimentelles Konzept zu Messung von $\sin^2\Theta_w$

Um den Weinbergwinkel zu messen, nutzt das P2-Experiment[j] die Beziehung des Winkels zur schwachen Ladung von Protonen[f].

$$Q_w(p) = 1 - 4 \cdot \sin^2(\Theta_w) \quad (2)$$

Hierbei ist  $Q_w(p)$  die schwache Ladung des Protons und  $\Theta_w$  der Weinbergwinkel. Um die schwache Ladung des Protons zu messen, nutzt man aus, dass die schwache Wechselwirkung paritätsverletzend ist. Dies bedeutet, dass man unter Raumspiegelung ein anderes Ergebnis für das gleiche Experiment erhält. Da man den Beschleuniger nicht einfach spiegeln kann, macht man sich einen Trick zunutze. Der Spin von Teilchen ändert sich nicht unter der Raumtransformation. Das bedeutet, dass durch Drehen des Spins in die entgegengesetzte Richtung der gleich Effekt erzielt wird, wie bei einer Spiegelung des Raumes.

Aus diesem Grund streut man mit dem polarisierten Elektronenstrahl von MESA[i, g] vollelastisch mit einem Wasserstoff-Target. Unter den beiden Spineinstellungen ergeben sich aufgrund der Paritätsverletzung zwei unterschiedliche Wirkungsquerschnitte. Diese Asymmetrie kann man nun messen. Sie ist direkt an die schwache Ladung des Targetteilchens gekoppelt:

$$A^{PV} = \frac{-G_F Q^2}{4\pi\sqrt{2}\alpha} \cdot (Q_w(p) - F(Q^2)) \quad (3)$$

$A^{PV}$  steht für die Paritätsverletzende Asymmetrie,  $G_F$  ist die Fermi Kopplungskonstante,  $Q^2$  der quadratische Impulsübertrag und  $\alpha$  die Kopplungskonstante der elektromagnetischen Wechselwirkung  $\frac{1}{137}$ .

Der Beitrag aus der Protonenstruktur  $F(Q^2)$ , der aus einer Addition verschiedener Formfaktoren wie elektrischer oder magnetischer Formfaktor besteht, kann mithilfe von Modellen bei dem entsprechenden Impulsübertrag berechnet oder gemessen werden.

Die Asymmetrie lässt sich über die folgende Beziehung messen:

$$A_{gem} = \frac{\sigma_{\uparrow} - \sigma_{\downarrow}}{\sigma_{\uparrow} + \sigma_{\downarrow}} \quad (4)$$

## 2. Das P2-Experiment

Die Wirkungsquerschnitte selbst lassen sich dann noch in Zählraten umschreiben. Diese Zählraten können nun direkt im Experiment gemessen werden.

$$A_{gem} = \frac{N_{\uparrow} - N_{\downarrow}}{N_{\uparrow} + N_{\downarrow}} \quad (5)$$

Im unten stehenden Bild sieht man nun das Design des P2-Experiments.

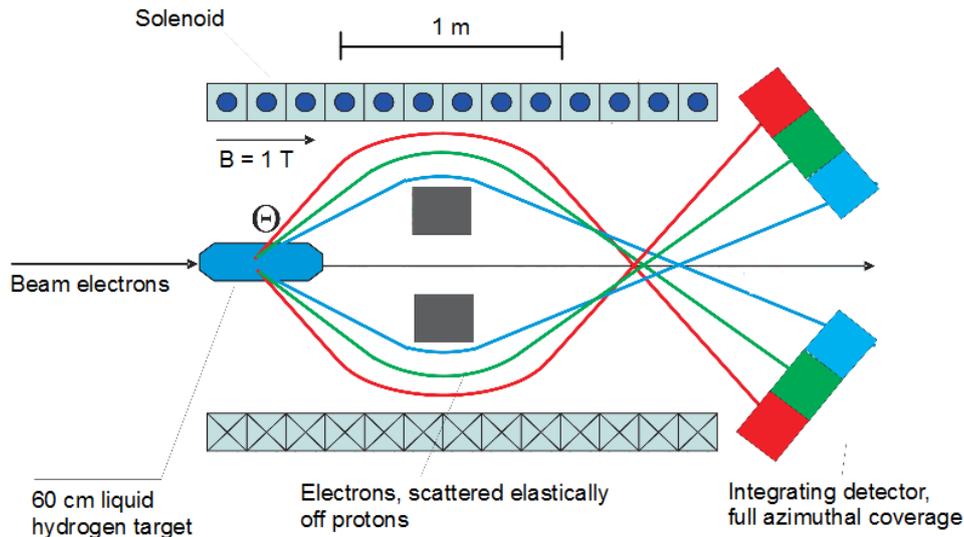


Abbildung 2: Schematische Zeichnung eines PMTs

Der Elektronenstrahl des Beschleunigers MESA trifft ein unpolarisiertes flüssiges Wasserstoff-Target. Die an den Atomkernen (Protonen) gestreuten Elektronen werden durch den Solenoiden abgelenkt. Dadurch findet eine Trennung der Signalelektronen zum Untergrund statt. Die abgelenkten Elektronen treffen dann auf einen Ring aus Quarz-Stäben welcher sich am hinteren Ende befindet. Der Ring deckt den vollen Azimutwinkel ab und akzeptiert Elektronen, die unter einem Polarwinkel von  $25\text{-}45^\circ$  am Target gestreut wurden. Der Elektronenstrahl ist polarisiert und ändert durch eine sogenannte Pockelzelle seine Helizität mit einer Umschaltfrequenz von bis zu maximal 10 kHz zwischen „+“ und „-“ bzw. „ $\uparrow$ “ und „ $\downarrow$ “. Jetzt kann man in dem jeweiligen Helizitätsfenster aufnehmen, wieviele Teilchen auf welchen Teil des Detektors getroffen sind und somit die Asymmetrie zwischen beiden Spineinstellungen aufnehmen.

Das Prinzip der Messung ist im Grundgedanke relativ simpel, aber das Problem steckt in der Umsetzung. Um die gewünschte Sensitivität für neue Physik jenseits des Standardmodells zu erreichen, soll die Asymmetrie mit einem Fehler von  $\Delta A^{exp} \leq 0.44 \text{ ppb}$  (parts per billion) bestimmt werden. Für die Asymmetrie selbst erwartet man einen Wert von  $A^{exp} \sim -28.35 \text{ ppb}$ . Das bedeutet einen relativen Fehler  $\leq 1.5\%$ .

Um dieses Ziel zu erreichen, müssen die statistischen und systematischen Fehlerquellen genau bekannt sein, und es muss sichergestellt werden, dass man diese hinreichend genau messen kann.

## 2. Das P2-Experiment

---

Der statistische Fehler bei solchen Zählexperimenten ist  $\frac{1}{\sqrt{N}} = \Delta A_{stat}$ , wobei  $N$  die Anzahl der Messpunkte ist (Dies gilt auch nur bei einem Polarisationsgrad des Strahls von 100%, sonst wird der Fehler größer). Dadurch, dass dieser  $< 0.44 \cdot 10^{-9}$  sein muss, ist eine hohe Luminosität erforderlich, um das Experiment innerhalb einer sinnvollen Zeitspanne beenden zu können. Deshalb ist ein Beschleuniger wie MESA mit einem hohen Strahlstrom wie  $150\mu A$  von Nöten. Zusätzlich wird auf ein langes (60cm) Flüssigwasserstoff-Target geschossen. Damit erreicht man eine Luminosität von  $L = 2.4 \cdot 10^{39} \text{cm}^{-2}\text{s}^{-1}$  und eine Rate von etwa 200 GHz, welche auf die Detektoren treffen. Trotz dieser extrem hohen Rate, ist eine Meßzeit von 10000h veranschlagt, um den gewünschten statistischen Fehler zu erreichen.

Es gibt auch einige systematische Fehlerquellen zu beachten, welche nicht mit der Zeit kleiner werden. Diese müssen so weit es geht unterdrückt werden oder gut genug bekannt sein, um sie später aus dem Messwert entfalten zu können. Dazu dienen Strahlstabilisierungen und Überprüfungen, da eine Schwankung des Strahls helizitätskorrelierte oder apparative Asymmetrien einführen, welche die richtige Asymmetrie überlagert. Desweiteren muss der Polarisationsgrad des Strahls sehr genau bekannt sein. Angestrebt ist ein relativer Fehler von  $\frac{\Delta P}{P} = 0.5\%$ .

Auch die gesamte Ausleseelektronik muss rauscharm sein, damit das Signal das sich so mühsam erarbeitet wurde mit genügend Aussagekraft digitalisiert werden kann. Die Elektronik muss den hohen Raten standhalten und dabei möglichst linear und rauscharm bleiben.

Diese Arbeit befasst sich mit den ersten Erfahrungen, die mit der Elektronik, welche an diesem neuen Experiment zum Einsatz kommen soll, gemacht wurde.

## 3. Photomultiplier Röhren und Spannungsteiler

### 3.1. Allgemeines

Geladene Teilchen können beim Durchgang durch Materie durch Szintillation und Cherenkov-Licht detektiert werden. Um das daraus entstehende Lichtsignal im Rechner digitalisieren zu können, benötigt man einen Verstärker, der aus der geringen Anzahl Photonen ein messbares elektrisches Signal liefert. Sogenannte Photomultiplier Röhren (engl. Photomultiplier Tube, Abk. PMT) erfüllen diesen Zweck sehr gut, da sie im Gegensatz zu Halbleiterdetektoren auch einzelne Photonen detektieren können [d, c, a]. Es gibt kommerziell erhältliche PMTs für Wellenlängen mit sensitivem Bereich von Infrarot bis Ultraviolett.

Die Außenhülle des PMTs (meistens Glas) ist vakuumiert, so dass Elektronen darin durch angelegte Spannungen effektiv beschleunigt werden können. Im Inneren sind die wichtigsten Komponenten: Die Photokathode und ein System aus Dynoden zur Vervielfachung der ausgelösten Elektronen. Damit erreicht man typischerweise eine Verstärkung von  $10^6$ - $10^7$ . Die Anzahl der Elektronen soll dabei möglichst linear proportional zur Anzahl der eingegangenen Photonen sein.

In der folgenden Abbildung ist das vereinfachte Schema eines PMTs aufgetragen.

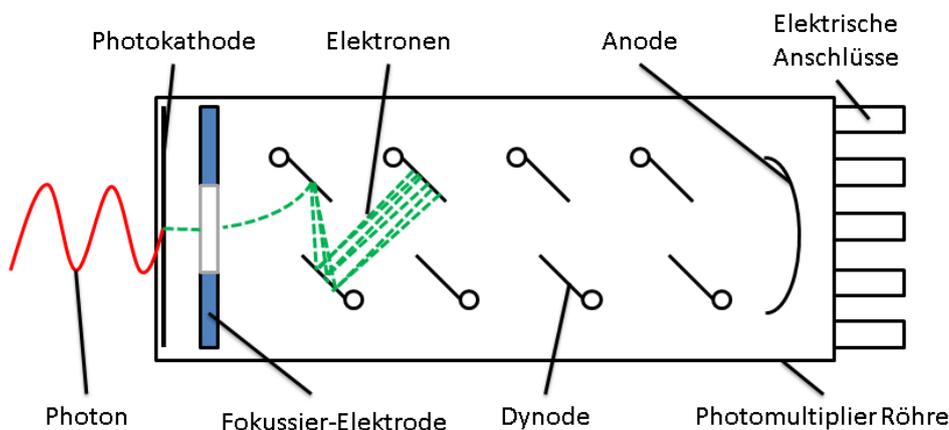


Abbildung 3: Schematische Zeichnung eines PMTs

### 3.2. Photoeffekt

Trifft ein Photon auf die photosensitive Kathode, so wird es mit einer gewissen Wahrscheinlichkeit in dieser dünnen Schicht absorbiert und sendet ein Elektron aus, welches die Energie

$$E = h\nu - \Phi \quad (6)$$

### 3. Photomultiplier Röhren und Spannungsteiler

hat.

Dabei ist  $E$  die kinetische Energie des Elektrons,  $h$  das Plancksche Wirkungsquantum,  $\nu$  die Frequenz des Lichts und  $\Phi$  die Austrittsarbeit des Kathodenmaterials. Damit ein Elektron emittiert wird, muss also die Energie des ankommenden Lichts einen bestimmten Wert übersteigen. Allerdings wird nicht jedes Photon, das ausreichend Energie zur Verfügung hat, von einem Elektron absorbiert, da die Schicht nur sehr dünn und deshalb semi-transparent ist.

Ein weiterer Faktor für eine gewünschte Absorptionsrate ist die Wahl des richtigen Kathodenmaterials. Je nach Wellenlänge der Photonen sind unterschiedliche Materialien geeignet.

Von der Austrittsarbeit hängt ab, wie stark das thermische Rauschen ist. Ist sie klein, so können thermische Elektronen im Leiterband des Kathodenmaterials energetisch genug sein und als Fehlpulse gezählt werden. Im Schnitt besitzen diese Elektronen bei Raumtemperatur zwar nur etwa 0.025 eV, aber da die Verteilung eine gewisse Breite hat, können die Elektronen an der oberen Grenze erheblich höhere Energien besitzen. Um nun die Effizienz einer Photokathode bei bestimmten Wellenlängen zu kennzeichnen, wurde die sog. Quanteneffizienz(QE),  $n(\lambda)$  definiert:

$$n(\lambda) = \frac{\text{Anzahl der emittierten Elektronen}}{\text{Anzahl der auf der Kathode aufgetroffenen Photonen}} \quad (7)$$

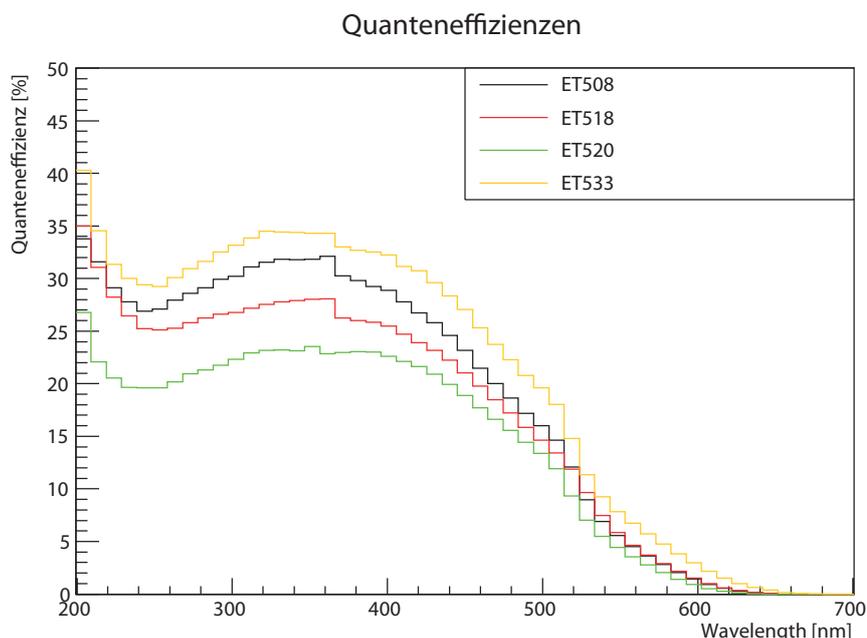


Abbildung 4: Quanteneffizienz als Funktion der Wellenlänge. Bei den Versuchen dieser Arbeit wurde ausschließlich der PMT von Electron Tubes mit der Seriennummer 518 verwendet.

Wobei  $\lambda$  die Wellenlänge des Lichts ist. Für eine ideale Photokathode wäre  $n(\lambda) = 100\%$ , aufgrund der vorher genannten Einschränkungen, wie z.B. ihrer Transparenz, erreichen reale

### 3. Photomultiplier Röhren und Spannungsteiler

---

Kathoden eine QE von etwa 20 – 30%.

#### 3.3. Verstärkung

Die Vervielfachung der Elektronen beruht auf dem Effekt der sekundär emittierten Elektronen. Das heißt: Aufgrund des elektrischen Feldes im PMT werden die Elektronen, die an der Photokathode ausgelöst wurden, beschleunigt und treffen, wie in Abb. 4 zu sehen, auf die erste Dynode. Da die Elektronen nun eine höhere Energie haben, lösen sie dort weitere Elektronen aus, proportional zu ihrer Energie. Von dort aus werden nun eine höhere Anzahl an Sekundärelektronen zur zweiten Dynode beschleunigt, an welcher der gleiche Prozess stattfindet. Der Faktor, um den eine einzelne Dynode die Anzahl der Elektronen verstärkt, ist dabei gegeben durch:

$$\delta = \frac{\text{Anzahl der emittierten sekundär Elektronen}}{\text{Anzahl der aufgetroffenen Elektronen}}. \quad (8)$$

Dieser Faktor sollte groß sein, um mit wenig Dynoden ein gut messbares elektrisches Signal zu erhalten. Hierbei werden für konventionelles Dynodenmaterial (BeO, MgO und Cs<sub>3</sub>Sb) ein maximaler Faktor von 10 erreicht (für hohe Spannungen zwischen den Dynoden), allerdings ist ein Faktor von 4-6 der Normalfall.

Man kann den Faktor außerdem erhöhen, indem man das Dynodenmaterial dotiert und somit die Potential-Barriere, die beim Übergang der Elektronen ins Vakuum besteht, absenkt. Man erreicht damit eine Verstärkung von ca. 50-60 pro Dynode.

Um nun eine Verstärkung von 10<sup>6</sup> zu erhalten, muss man wie in Abbildung 4 zu sehen mehrere Dynoden hintereinander schalten. Dazu muss die folgende Dynode immer auf einem höheren positiven Potential liegen als die vorige. Da die ausgelösten Sekundärelektronen eine sehr niedrige Energie haben, ist es sehr einfach, diese auf die nächste Dynode zu leiten. Es ergibt sich somit eine Verstärkung bei  $N$  Dynoden von:

$$A = \alpha \cdot \delta^N. \quad (9)$$

Dabei ist  $A$  die Gesamtverstärkung und  $\alpha$  der Anteil der Photoelektronen, die von den Dynoden verstärkt werden. Somit kommt man bei zehn Dynoden mit einer durchschnittlichen Vervielfachung von  $\delta = 5$  und  $\alpha \approx 1$  auf eine Verstärkung von 5<sup>10</sup> was in der Größenordnung von 10<sup>7</sup> liegt. Benutzt man nun dotiertes Dynodenmaterial, benötigt man für eine ähnliche Verstärkung nur etwa 5 Dynoden.

Wäre  $\delta$  eine feste Zahl, so hätte jeder Puls, der von einem einzelnen Photoelektron kommt, die gleiche Form und Höhe. Da die Sekundäremission aber ein statistisch verteilter Prozess ist gibt es eine Schwankung um  $\delta$  herum. Als einfachstes Modell kann nun angenommen werden, dass diese Schwankung einer Poissonverteilung entspricht. Damit ergibt sich für den Mittelwert  $\delta$  eine Standardabweichung von  $\sqrt{\delta}$  und eine Varianz von  $(\frac{\sigma}{\delta})^2 = \frac{1}{\delta}$ .

Betrachtet man dies nun für  $N$  Dynoden, so folgt für die gesamte relative Varianz aus der Poissonstatistik:

$$\frac{1}{\delta} + \frac{1}{\delta^2} + \frac{1}{\delta^3} + \dots + \frac{1}{\delta^N} \approx \frac{1}{1 - \delta} \quad (10)$$

### 3. Photomultiplier Röhren und Spannungsteiler

---

Man sieht, dass für  $\delta \gg 1$  die Varianz, also die Streuung in der Form der aufgenommenen elektrischen Pulse, stark von der Schwankung der Verstärkung der ersten Dynode abhängt, bei der noch am wenigsten Elektronen involviert sind. Außerdem wird klar, je stärker die Verstärkung der einzelnen Dynoden, desto schärfer ist die Verteilung um  $\delta$ .

Mit einem kleinen Test kann festgestellt werden, ob die Verstärkung für verschiedene Versorgungsspannungen richtig gemessen wurde.

Wir wissen, dass  $\delta$  eine Funktion der Spannungsverteilung zwischen den Dynoden ist. Je höher die Spannung, desto höher ist auch  $\delta$ . Zusätzlich ist  $\delta$  abhängig von einem materialspezifischen Koeffizienten  $k$ . Somit kann man  $\delta$  wie folgt definieren:

$$\delta = a \cdot E(U)^k \quad (11)$$

wobei  $a$  hier nur eine Konstante und  $E(U)$  eine Funktion der Spannung ist. Man kann  $\delta$  auch über den Strom, der fließt, definieren.

$$\delta_n = \frac{I_{dn}}{I_{d(n-1)}} \quad (12)$$

Hier ist  $I_{dn}$  der Strom, der von der  $n$ -ten Dynode abfließt und  $I_{d(n-1)}$  der Strom, der von der vorigen Dynode zufließt. Somit kann man den Anodenstrom zusammenfassen zu:

$$I_A = I_k \cdot \alpha \cdot \delta_1 \cdot \delta_2 \cdots \delta_n. \quad (13)$$

mit  $I_k$  als Kathodenstrom. Teilt man nun durch  $I_k$ , so bekommt man die Verstärkung (Gain) des PMTs. Geht man nun von einem Photomultiplier aus bei dem  $\alpha = 1$  und die Spannung zwischen den Dynoden immer den selben Wert hat, so ergibt sich die Verstärkung  $\mu = \frac{I_A}{I_K}$  zu

$$\mu = (a \cdot E^k)^n = a^n \left( \frac{V}{n+1} \right)^{kn} = A \cdot V^{kn}, \quad (14)$$

wobei  $V$  die Betriebsspannung ist. Hier sieht man, dass der Zusammenhang für einen doppelloarithmischen Plot der Verstärkung gegen die Betriebsspannung linear sein sollte.

#### 3.4. Rauschen und Dunkelpulse

Die häufigste Ursache für ein statistisches Rauschen des PMTs sind die Elektronen die thermisch aus der Photokathode ausgelöst werden. Dabei lösen sich meist Einzel-Elektronen, die dann auch eine entsprechende Elektronen-Lawine auslösen.

Die Anzahl der thermisch ausgelösten Elektronen ist proportional zur Fläche der PMT-Kathode. Deshalb sollte man den, für die entsprechende Applikation, kleinstmöglichen PMT benutzen. Weiterhin kann man, wie der Name schon vermuten lässt, die Rate der thermisch ausgelösten Elektronen deutlich senken, indem der PMT gekühlt wird.

PMTs sollte man möglichst im Dunkeln lagern, auch wenn keine Spannung angelegt ist. Die

### 3. Photomultiplier Röhren und Spannungsteiler

---

Rate der Dunkelpulse steigt, einmal kurz dem Licht ausgesetzt, drastisch an und der PMT muss erst wieder für längere Zeit im Dunkeln liegen, bevor er erneut für die Detektion von kleinen Lichtmengen einsatzbereit ist. Setzt man den PMT mit eingeschalteter Spannung normalem Tageslicht aus, kann dieser durch die hohe Lichtmenge zerstört werden.

Ein weiterer Grund für Dunkelpulse ist die Radioaktivität der Stoffe innerhalb des Photomultipliers. Dabei ist der Hauptanteil des aktiven Materials im Glasgehäuse in Form von  $^{40}\text{K}$  zu finden. Durch  $\beta$ -Zerfall der K-Atome, wird ein kurzer Puls an Tscherenkow-Licht erzeugt, das wiederum Elektronen aus der Kathode auslösen kann. Aus diesem Grund, gibt es, falls die Untergrundstrahlung besonders niedrig sein soll, PMTs aus sehr schwach radioaktivem Material. Zusätzlich können andere externe Strahlungsquellen, Szintillations- oder Tscherenkow-Licht im Glas erzeugen, wie z.B. den Myonen aus den Luftschauern.

Ein letzter aber wichtiger Punkt in der Betrachtung des Rauschens ist die Entstehung von Nachpulsen, die, wie der Name schon sagt, aufgrund des wirklichen Pulses, nach diesem auftreten. Es gibt zwei Hauptgründe für Nachpulse. Der erste ist, dass innerhalb des Vervielfachungssystems Photonen entstehen, die ihren Weg zurück zur Kathode finden, um dort erneut ein Elektron auszulösen und somit ein zweiter Puls entsteht. Dabei entstehen meistens nur sehr wenige Photonen, so dass der Nachpuls nur sehr klein ist und dem eines Single Photons entspricht. Der zweite Grund für diese Pulse ist ein schlechtes Vakuum in der Röhre. Das Restgas kann dann von den Elektronen ionisiert werden. Die positiv geladenen Ionen driften dann zur Kathode hin. Da die Ionen des Gases viele Photoelektronen auslösen, entsteht dadurch ein sehr großer Nachpuls.

#### 3.5. Spannungsteiler

Um einen PMT zu betreiben, muss man ihn an eine Spannungsquelle anschließen. Hier muss sichergestellt werden, dass jede Dynode auf das richtige Potential gebracht wird. Dabei ist die einfache Regel: Jedes Elektron muss beschleunigt werden, also muss die erste Dynode auf einem positiveren Potential sein als die Kathode und jede weitere Dynode muss auf einem höheren Potential sein als die vorige.

Diese Versorgung kann man nun durch individuelle Spannungsquellen erreichen, die jede Dynode auf ihr entsprechendes Potential bringen.

Allerdings verwendet man im Großteil aller Anwendungen einen Spannungsteiler, um den gewünschten Effekt zu erreichen. Herkömmliche Spannungsteiler bestehen meist nur aus einem einfachen Spannungsteiler mit Anschluss für Hochspannung und Kondensatoren an den letzten 2-3 Dynoden. Das Schema eines solchen Teilers sieht man in Abbildung 5 aufgezeichnet.

### 3. Photomultiplier Röhren und Spannungsteiler

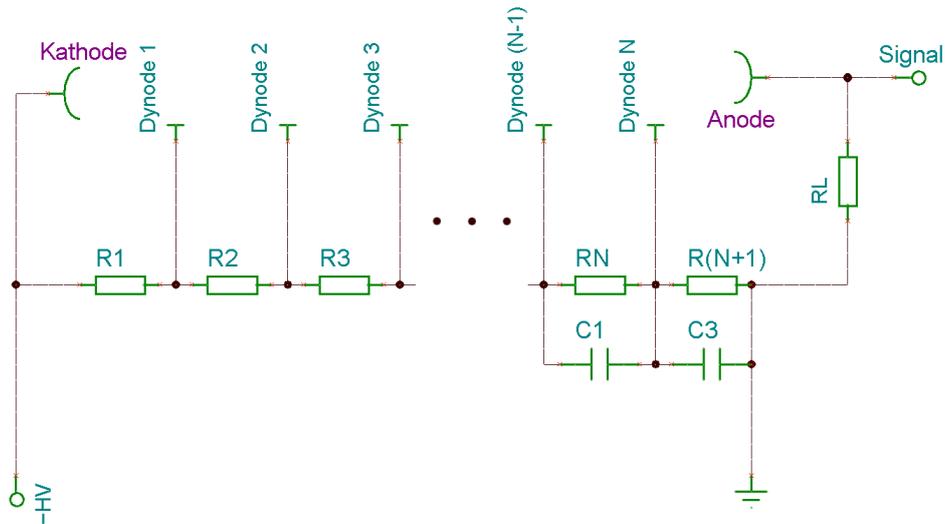


Abbildung 5: Schema eines passiven Spannungsteilers für einen PMT

Um die Verstärkung eines PMTs konstant zu halten, muss man sicherstellen, dass die Spannung an jeder einzelnen Dynode konstant bleibt. Damit dies der Fall ist, muss der Strom, der über den Spannungsteiler fließt, etwa 100 mal so hoch sein wie der Strom, der an der jeweiligen Dynode abfließt. Dies ist vor allem am Ende des Spannungsteilers ein Problem, da hier extrem viele Elektronen in kurzer Zeit abfließen. Um diese kurzen Stromspitzen, welche zu Spannungsabfällen führen würden, abzufedern, kommen Kondensatoren zum Einsatz. Diese entladen sich bei der punktuellen Belastung, und laden sich zwischen zwei Kaskaden wieder auf.

Es handelt sich hierbei um einen sogenannten passiven Spannungsteiler, da dort nur passive Elemente wie Widerstände und Kondensatoren verbaut werden. Da ein solcher Spannungsteiler nur aus wenigen Elementen besteht, kann die Platine, auf der die Schaltung sitzt, sehr klein sein und nimmt somit nicht viel Platz weg, was bei einem Detektor der aus mehreren hundert Modulen besteht, ein nicht unerheblicher Vorteil ist. Zudem sind die Bauteile sehr billig, so dass die Materialkosten der Bauteile selbst für eine hohe Modulanzahl vernachlässigbar ist. Der Nachteil bei passiven Spannungsteilern ist, dass bei höheren Raten keine Zeit bleibt, um die Kondensatoren zwischen zwei Pulsen wieder aufzuladen. Das bedeutet, dass die Verstärkung eines PMTs der mit passiven Spannungsteilern betrieben wird, schon bei Raten von 1 MHz erheblich absinkt. Da im P2-Experiment mit Raten in der Größenordnung 200 GHz zu rechnen ist (was bei 200 Modulen dann also etwa 1 GHz pro PMT ausmacht), wird diese Art von Spannungsteiler für das P2-Experiment nicht in Frage kommen.

Um auch bei höheren Raten die Spannungsfestigkeit zu garantieren, verbaut man auf dem Spannungsteiler einige aktive Elemente, die Spannung an den einzelnen Dynoden regeln. Einen solchen Spannungsteiler sieht man in Abbildung 6.

### 3. Photomultiplier Röhren und Spannungsteiler

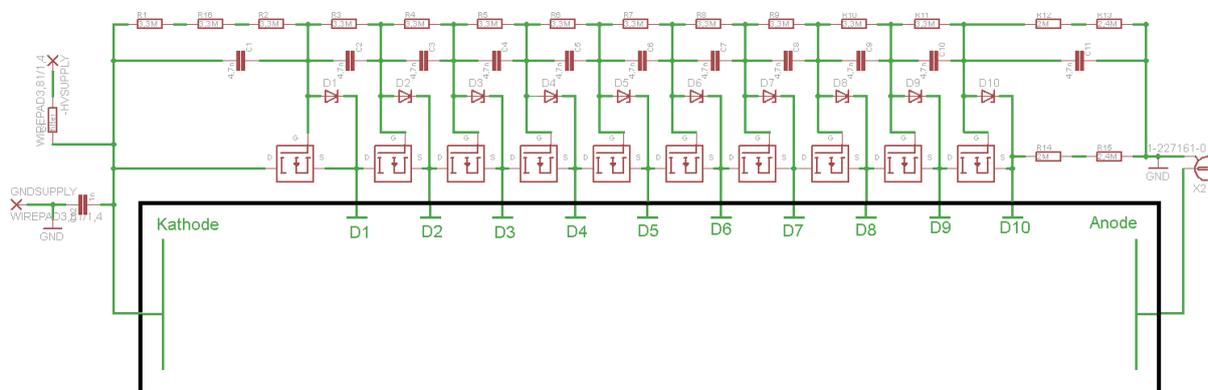


Abbildung 6: Schema eines aktiven Spannungsteilers

Wie in Abbildung 6 zu sehen, gibt es auch hier eine Spannungsteilerkette. Die Kondensatoren gibt es nun schon ab dem ersten Glied, nicht nur für die letzten. Darunter sitzt aber nun der aktive Part, ein Transistor und eine Diode für jedes Dynodenglied. Es handelt sich bei den Transistoren um N-Kanal Mosfets und bei den Dioden um Zehnerdioden.

Mosfets sind sogenannte Feldeffekttransistoren. Im Gegensatz zu normalen Transistoren nennt man hier die Anschlüsse nicht Basis, Collector und Emitter, sondern Gate, Drain und Source. Mithilfe eines Transistors lassen sich Ströme oder Spannungen steuern. Bei normalen Transistoren steuert man das im Normalfall über die Basis, bei Mosfets über das Gate. Erfüllt man die entsprechende Bedingung, welche je nach Ausführung des Transistors verschieden ist, so wird die Verbindung Collector-Emitter oder in unserem Fall Drain-Source leitend oder sperrend, je nachdem welcher Zustand der Ursprungszustand ist.

Bei der Ausführung N-Kanal handelt es sich um einen Mosfet, bei dem die Verbindung Drain zu Source nur dann leitend ist, wenn das Gate auf einem niedrigeren Potential liegt als der Source Anschluss. Ist dies nicht der Fall, so ist die Verbindung sperrend.

Bei Zehnerdioden handelt es sich um spezielle Dioden, die eigentlich ausschließlich in Sperrrichtung betrieben werden. Sie funktionieren wie normale Dioden in Sperrrichtung, bis ihre Zehnerspannung erreicht wurde, danach werden sie quasi normal leitend.

Um das Ganze etwas besser sichtbar zu machen, ist in Abbildung 7 nochmal der Teil zwischen Kathode und Dynode 2 größer aufgetragen.

### 3. Photomultiplier Röhren und Spannungsteiler

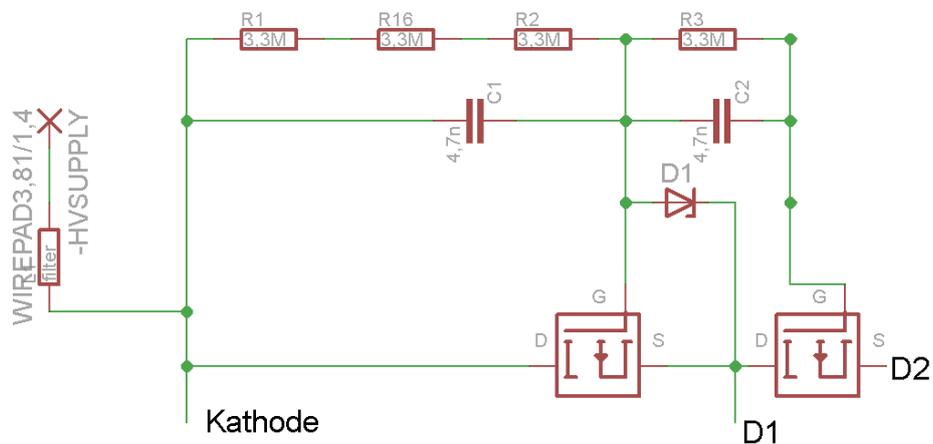


Abbildung 7: Erstes und zweites Glied der Teilerkette

Die aktive Regelung funktioniert wie folgt:

Im Ausgangszustand ist der Transistor gesperrt. Er wird immer dann leitend, wenn die Spannung am Source höher ist als die am Gate. Im Ausgangszustand haben Gate und Source das gleiche Potential, der Drain liegt auf einem niedrigen. Kommt nun ein Puls, so fließen Elektronen aus der jeweiligen Dynode (hier Dynode 1), und die Spannung steigt am Source. Aufgrund der Diode kann der Strom nun aber nicht vom Gate aus nachfließen und somit nicht durch die Teilerkette, sodass dort die Spannungen stabil bleiben. Da der Source jetzt aber auf einem höheren Potential liegt als das Gate, wird der Transistor leitend und der Strom fließt direkt von der Spannungsversorgung zu diesem Punkt. Dies passiert, bis wieder die alte Spannung erreicht wurde. Dieses Prinzip funktioniert durchgängig in jedem Teil der Kette gleich. Also sind unter Last alle Dynoden direkt mit der Hochspannungsquelle verbunden. Dadurch sind höhere Raten ohne Verstärkungseinbrüche möglich [1].

#### 3.5.1. Spannungsteiler für das P2-Experiment

Für das P2-Experiment sollen aktive Spannungsteiler verwendet werden, allerdings werden diese noch einmal modifiziert. Beim späteren Experiment soll in regelmäßigen Abständen ein sogenanntes Tracking der Teilchen durchgeführt werden, bei dem die Trajektorien einzelner Teilchen zwischen Target und Detektor bestimmt werden. Dies dient dazu, die Impulsüberträge  $Q^2$  mit hoher Genauigkeit zu messen. Beim Tracking wird mit einer sehr viel niedrigeren Rate auf das Target geschossen, da sich nur so einzelne Teilchenbahnen nachverfolgen lassen. Um das daraus erfolgende Signal ausreichend genau auslesen zu können, benötigt man PMTs mit einer hohen Verstärkung ( $\sim 10^6 - 10^7$ ). Während des normalen Experiments hingegen sind die Raten so hoch, dass man die Photomultiplier nicht mehr bei so hohen Verstärkungsfaktoren betreiben kann. Also muss die Verstärkung für die beiden Betriebsmodi geändert werden. Um die Verstärkung eines PMTs zu ändern gibt es 3 Möglichkeiten:

- Verstimmen der Spannungsteilerkette
- Änderung der Versorgungsspannung
- Kurzschließen der Dynoden

Der Hersteller eines PMTs liefert im Normalfall einen Schaltplan für einen Spannungsteiler mit, der für die jeweilige Röhre, für verschiedene Anforderungen optimiert ist. Ändert man die Verhältnis der Widerstände an den einzelnen Stufen, so kann man damit leichte Änderungen in der Gesamtverstärkung des PMTs erzielen. Dies hat allerdings einige Nachteile. Da diese Änderungen in die Sammeloptik des PMTs eingreift, von der nur der Hersteller Genaueres weiss, kann man nicht genau sagen, ob der PMT noch so funktioniert, wie vom Hersteller angegeben. Die Effekte sind nicht in der Größenordnung in der man im P2-Experiment umschalten möchte. Die Änderung von einzelnen Widerständen während des Betriebs ist aufwendig, vor allem bei sehr vielen Modulen. Aus diesem Grund ist die erste Methode nicht geeignet.

Die einfachste Methode stellt die Änderung der Versorgungsspannung da. Leider ändert sich damit auch die Beschleunigungsspannung zwischen den einzelnen Dynoden. Allerdings soll die Änderung in der Verstärkung so groß sein, dass man den PMT deutlich unter 1000 Volt Hochspannung betreiben müsste. Röhren mit einer hohen Verstärkung, wie sie für das Tracking notwendig sind, sind für höhere Spannungen ausgelegt, und funktionieren auch nur in ihrem Arbeitsbereich ideal. Es wäre deswegen besser, wenn man die Hochspannung zur Versorgung nicht zu klein wählen müsste, um die gewünschte Verstärkung zu erreichen.

Die letzte Methode ist das Kurzschließen mehrerer Dynoden mit der Anode. Somit koppelt man das Signal schon an einer früheren Dynode aus, was dazu führt, dass der Verstärkungsfaktor für diese wegfällt. Dies ermöglicht eine Veränderung der Verstärkung, ohne aber den Spannungsabfall zwischen den einzelnen Dynoden zu ändern. Somit bleibt der PMT beim Betrieb in dem vom Hersteller empfohlenen Arbeitsbereich.

Es wurde hier für die letzte Methode entschieden, und der finale Schaltplan für den Spannungsteiler ist auf zwei Platinen umgesetzt worden, welche durch einen Steckverbinder verbunden sind. Die finalen Schaltpläne sind in Abbildung 8 abgebildet.

### 3. Photomultiplier Röhren und Spannungsteiler

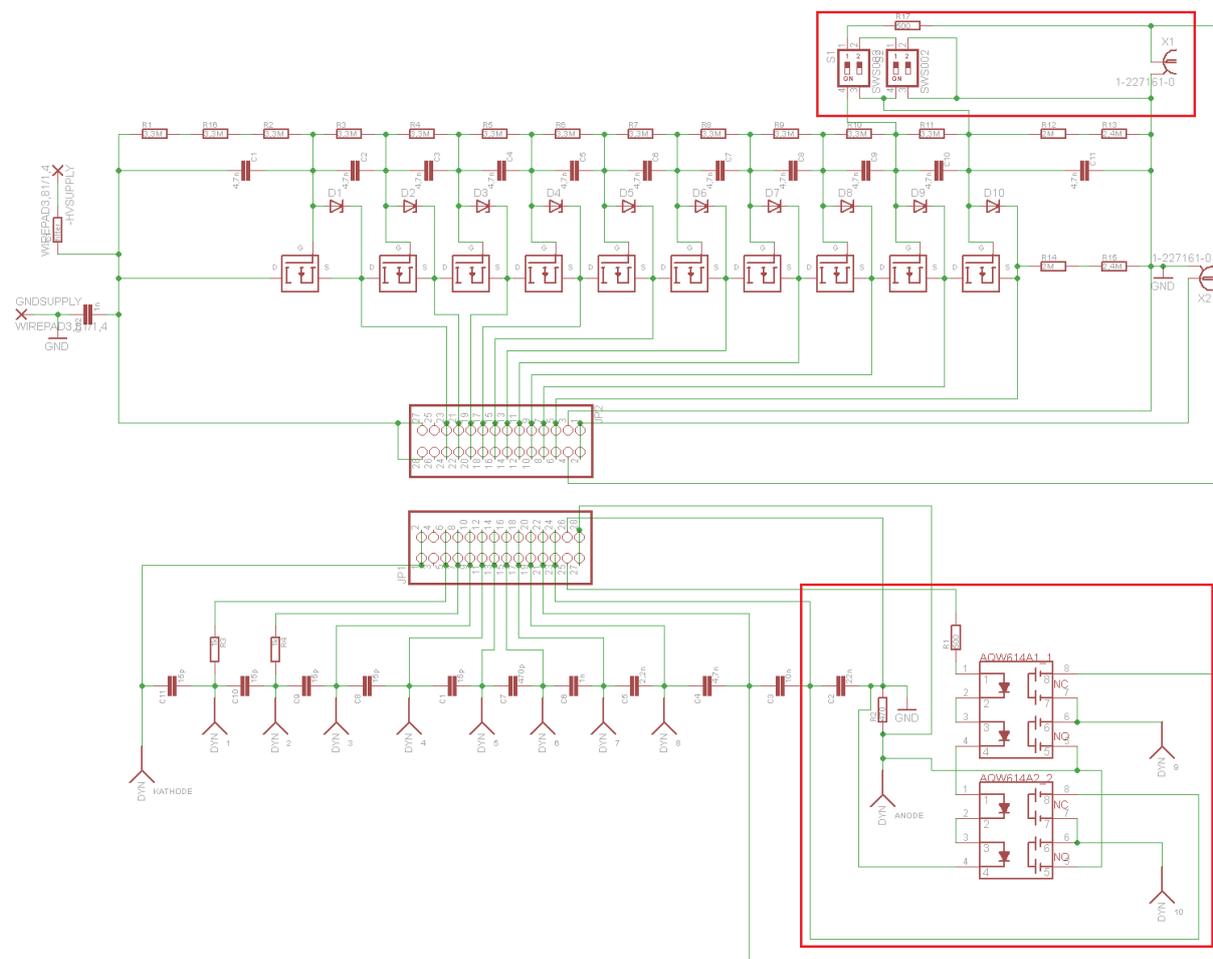


Abbildung 8: Finaler Schaltplan des Spannungsteiler-Prototyps

In den roten Kästen ist der Teil markiert, wo die Dynoden überbrückt werden. Es kommen hierbei sogenannte Photomos-Relais zum Einsatz. Wie bei normalen Relais handelt es sich um elektrisch bedienbare Schalter, allerdings benötigen Photomos-Relais viel kleinere Ströme, um zu schalten, und sind von der Baugröße her so klein, dass sie auf den Platinen gut Platz finden. Dies ist für einen einzelnen Prototyp nicht entscheidend, später beim gesamten Detektor mit mehreren hundert Modulen und begrenztem Platz ein wichtiger Punkt.

Mithilfe der insgesamt vier Relais werden die letzten zwei Dynoden mit der Anode kurzgeschlossen. Dies wird über eine BNC Buchse gesteuert. Schließt man an diese +5 Volt an, schalten die Relais. Da man hier den Spannungsteiler um zwei Stufen verkürzt, teilt man die Gesamtspannung nun nicht mehr auf insgesamt zwölf, sondern nur noch zehn Stufen auf. Dies ändert allerdings die Spannungen zwischen den einzelnen Dynoden, was den gleichen Effekt hätte wie die Änderung der Gesamtspannung. Um dies zu umgehen, muss die Gesamtspannung verringert werden, wenn die Relais schalten. Es wurde darauf geachtet, dass die Spannung zwischen der Kathode und der ersten Dynode konstant bleibt, da der Verstärkungsprozess eines PMTs, wie schon in Kapitel 3.3 beschrieben, maßgeblich von der ersten Stufe abhängt.

### 3. Photomultiplier Röhren und Spannungsteiler

Im Betrieb mit allen zwölf Dynoden fallen zwischen Kathode und Dynode 1, 22,5% der Gesamtspannung ab. Schaltet man die Relais und betreibt den Spannungsteiler damit also effektiv mit 10 Dynoden, so erhöht sich dieser Wert auf 27,5%. Also ist der neue zu wählende Wert der Spannung:  $U_{ein} = \frac{0,225}{0,275} \cdot U_{aus}$ .

In einem späteren Kapitel werden die Methode und Messungen vorgestellt, um den Faktor der Verstärkungsänderung bestimmen.

Die beiden Platinen wurden so gestaltet, dass sie nicht über den Sockel des PMTs hinausstehen, damit die PMTs, wenn sie nebeneinander im Detektor sitzen, nicht aufgrund ihres Spannungsteilers gegeneinander stoßen. Um das Platinen Layout zu erstellen wurde EAGLE benutzt und der Prototyp in der Elektrowerkstatt der KPH Uni Mainz erstellt. Bei EAGLE handelt es sich um eine Elektronik Layouter-Software, mit der sich Schaltpläne auf ein Platinen Layout übertragen lassen. In Abbildung 9 sieht man das Layout der beiden Platinen.

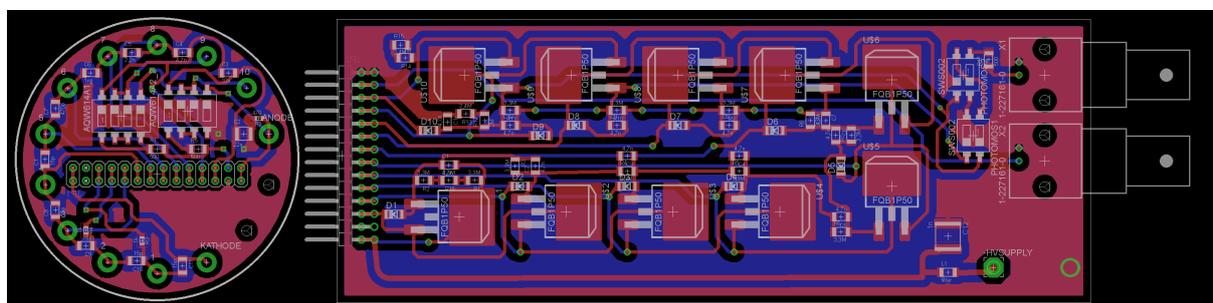


Abbildung 9: Layout des Spannungsteiler-Prototyps

Man sieht hier auf dem rechten Bild die drei Anschlüsse für die Relais (oben), das Signal (Mitte) und die Hochspannungsversorgung (unten). Die Hochspannungsversorgung ist hier nur durch Lötpad realisiert, auf dem ein Kabel oder Stecker direkt aufgelötet werden kann. Dies ist nur eine provisorische Lösung für einen Prototyp und sollte später durch eine dauerhafte Lösung mit Abschirmung der Hochspannungsleitung und Zugentlastung ersetzt werden.

In Abbildung 10 ist ein Foto der beiden zusammengesteckten Platinen zu sehen, welche schon auf den Sockel für den PMT montiert wurde, sodass man diesen nur noch auf den PMT aufschieben muss.



Abbildung 10: Foto des Gesamtaufbaus des Spannungsteiler-Prototyps

### 4. Analog Digital Wandler

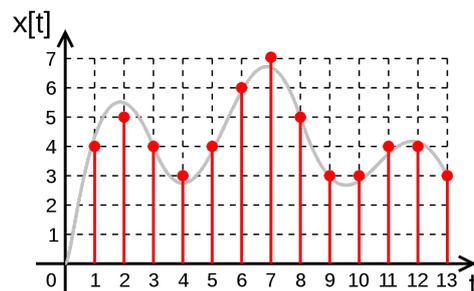
Ein Analog-Digital-Wandler (engl. Analog-to-Digital Converter, Abk. ADC), wandelt wie der Name schon sagt ein analoges in ein digitales Signal um [1, k].

Die für uns wichtige Anwendung ist die Messwerterfassung in Experimenten und Forschungsanlagen. Wie im vorherigen Kapitel beschrieben, erfasst ein PMT kleine Lichtsignale und verstärkt es. Nun hat man ein messbares Signal, dieses ist aber immer noch analog und kann so nur über Analog-Elektronik weiteranalysiert werden. Hilfreich ist es, dieses Signal zu digitalisieren, so kann es einfach am Rechner über Algorithmen weiterverarbeitet werden.

#### 4.1. Funktionsweise

Das ADC wandelt das analoge zeitlich kontinuierliche Eingangssignal in diskrete Werte um. Diese Werte werden dann in eine Kanalnummer des ADCs eingeordnet. Das Signal wurde also digitalisiert. Da man nur endlich viele Kanäle zur Verfügung hat, erfolgt dabei immer eine Quantisierung.

In Abbildung 11 kann man sich das Ganze noch einmal grafisch veranschaulichen. Auf der Y-Achse sieht man die einzelnen diskreten Kanalnummern des ADCs aufgetragen. Das analoge Signal kann nur auf genau diese Kanäle aufgeteilt werden, es gehen also Informationen über das Signal verloren. Ab wann dieser Verlust nicht mehr vernachlässigbar klein ist, kommt auf die Anforderung an das Signal an.



Quelle: <https://de.wikipedia.org/wiki/Analog-Digital-Umsetzer>

Abbildung 11: Quantisierung eines analog Signals (grau) in digitale Werte (rote Punkte)

Die Hauptparameter eines ADCs kann man gut anhand des Bildes sehen. Erstens, wie viele solcher Kanalnummern gibt es, dies nennt man die Bittiefe oder die Bitanzahl des ADCs. Dies liegt am Binärsystem, in dem die Anzahl der zu Verfügung stehenden Kanäle gemessen wird. Ein ADC mit einer Bittiefe von 12 hat also 4096 Kanäle um ein Signal zu digitalisieren.

Der zweite Hauptparameter ist die Abtastrate, also wie oft pro Sekunde ein Wert aus dem Zeitsignal genommen wird. Es gibt auch spezielle ADCs, die keine eigentliche Abtastrate besitzen, sondern über einen gewissen Zeitraum die Ladung des Signals erfassen, und dann für diesen Zeitraum einen Wert ausgeben.

Es geht in dieser Einleitung hauptsächlich um den ersten Typ von ADCs, da die beiden analog zu einander funktionieren. Der eine ADC wiederholt den Konversionsvorgang lediglich periodisch,

## 4. Analog Digital Wandler

---

während der andere für einen selbst gewählten Zeitraum das Signal akkumuliert und dann umwandelt.

### 4.1.1. Abtastung des Signals

Um ein Zeitsignal verlustfrei zu diskretisieren, ist die Bandbreite des Eingangssignals entscheidend. Das Nyquist-Shannon-Abtasttheorem besagt, dass die Abtastrate mindestens doppelt so groß sein muss wie die höchstmögliche Frequenz des Eingangssignals. Tut man dies nicht, ist es möglich, dass im rekonstruierten Signal Frequenzen enthalten sind, die nicht aus dem Eingangssignal stammen.

Es ist deshalb wichtig, dass das Eingangssignal bandbegrenzt wird. Dies geschieht meist durch Tiefpassfilter oder Integratoren (diese integrieren alle Frequenzen über der Grenzfrequenz zu einem Gleichstrom-/Gleichspannungsanteil auf). Ist das Signal jedoch zu hochfrequent, um dieses Theorem einzuhalten und kann nicht Bandbreiten-begrenzt werden, gibt es, falls das Signal periodisch ist, noch die Möglichkeit der Mehrfachabtastung. Dies wird aber an dieser Stelle nicht ausführlicher diskutiert, da es in dieser Arbeit nicht zur Anwendung kommt.

Während der Signalumsetzung muss das Eingangssignal bei den meisten Umwandelverfahren kontant bleiben. Dies geschieht meist durch sogenannte Sample-and-Hold Schaltungen. Je nachdem wie lange das Verfahren zur Umwandlung dauert, begrenzt dies die maximale Abtastrate des ADCs. Die Abtastrate ist in der Realität nicht komplett konstant, diesen Effekt nennt man Jitter. Dieser Fehler führt bei der Rekonstruktion des Signals zu Fehlern, da die aufgenommenen Werte nicht äquidistant sind.

### 4.1.2. Quantisierung

Wie schon einmal erwähnt, hat ein ADC nur endlich viele Kanäle, was zu einer Quantisierung des Signals führt. Die Abweichung von wahren Signal zur quantisierten Kanalnummer des ADCs, nennt man Quantisierungsfehler. Je mehr Bit (also Kanalnummern) ein ADC besitzt, desto kleiner wird dieser Fehler.

Durch sogenanntes Oversampling, kann die Anzahl der effektiven Bits die zum Auflösen des Signals vorhanden sind auch gesteigert werden. Oversampling bedeutet, dass die Abtastrate höher ist, als die vom Abtasttheorem geforderte für die jeweilige Anwendung. Somit kann man Mittelwerte zwischen den Messwerten bilden, was, wenn man das Abtasttheorem dadurch nicht verletzt, effektiv mehr Kanäle bringt da man nun auch halbe, viertel, etc. Kanalnummern zur Verfügung hat. In der Praxis addiert man die Kanalnummern mehrerer Samples (Einzelmessungen) einfach auf, was die maximal mögliche Kanalnummer erhöht, dafür kann weiterhin mit ganzen Zahlen gearbeitet werden.

Bei einem Oversampling von Faktor  $N$  erhält das ADC zusätzlich eine Bittiefe von  $\log_2(N)$ .

### 4.1.3. Quantisierungskennlinie

Ein realer ADC hat immer einen nichtlinearen Zusammenhang zwischen Eingangs- und Ausgangsgröße. Ist dieser jedoch sehr gering, also steigt beim Erhöhen der Eingangsspannung auch die Ausgangsgröße in konstanten Abständen, so spricht man trotzdem von einem linearen ADC. Um später einen großen Bereich des ADCs nutzen zu können, ist es deshalb wichtig zu überprüfen, ob ein ausreichend guter linearer Zusammenhang gegeben ist.

### 4.2. Abweichungen zur idealen Umwandlung

Neben dem durch die Bitzahl begrenztem unvermeidlichen Quantisierungsfehler gibt es weitere Abweichungen des Ausgangssignals, welche durch den ADC eingeführt werden.

#### 4.2.1. Nullpunktfehler

Im Datenblatt eines jeden ADCs wird ein Kanal angegeben, bei welchem die angebliche Null des ADCs liegt. Dies ist meistens nicht der nullte Kanal, da man sonst etwas von Signalen, die nahe an der Null liegen, abschneiden würde. Meist ist auf diesem angegebenen Kanal noch ein Offset. Also existiert ein Nullpunktfehler. Diesen Fehler vermeidet man, indem man die Null seines ADCs möglichst mitmisst. Dies geschieht, indem man auch Daten zu einem Zeitpunkt aufzeichnet, bei welchem sicher kein Signal anliegt. So kann man nicht nur um diesen Nullpunkt korrigieren, sondern direkt auch noch das Eigenrauschen des ADCs bzw. das Rauschen der Messkette bis zum ADC ohne Last messen.

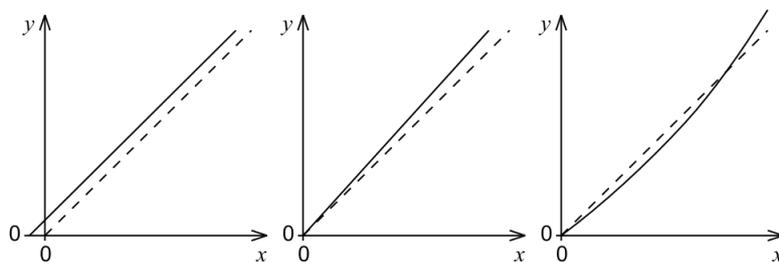
#### 4.2.2. Verstärkungsfehler

Man kann aus dem Datenblatt den Wert eines ADC-Kanals lesen. Man weiss also, wenn man sein Ausgangssignal um einen bestimmten Wert steigert, so muss auch das ADC um einen konstanten Wert steigern. Erhöht man nun sein Ausgangssignal wieder um besagten Wert, darf sich auch die Ausgabe des ADCs wieder nur um den konstanten Wert erhöhen. Ist dies nicht der Fall, so liegt ein Verstärkungsfehler vor. Ist dieser multiplikativ, so ist er von linearer Natur, und kann leicht korrigiert werden.

#### 4.2.3. Nichtlineare Fehler

Ist wie im letzten Abschnitt schon erwähnt der Zusammenhang zwischen Eingabe und Ausgabe nicht ausreichend linear, treten bei Nutzung eines großen Bereichs Fehler durch Nichtlinearitäten auf, welche nicht auf einfache Weise zu korrigieren sind.

In Abbildung 13 sind die drei verschiedenen Arten dieser Fehler noch einmal aufgetragen.



Quelle: <https://de.wikipedia.org/wiki/Analog-Digital-Umsetzer>

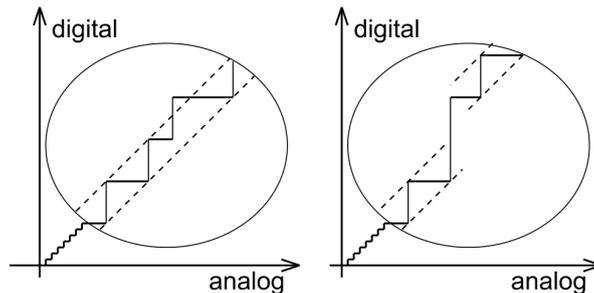
Abbildung 12: Mögliche Abweichung realer von idealen ADCs, Nullpunktfehler(links), Verstärkungsfehler(mitte), nicht lineare Fehler(rechts)

## 4. Analog Digital Wandler

---

### 4.2.4. Stufenfehler

Da die Ausgangswerte quantisiert sind, gibt es für einen Kanal einen kleinen Bereich von Eingangswerten, für den der ADC immer die gleiche Kanalnummer ausgibt. Dieser Bereich kann jedoch unterschiedlich breit ausfallen für jede Stufe. Ist dem so, so spricht man von einem Fehler in der Stufung. Es kann zudem auch passieren, dass ein Kanal nicht oder kaum getroffen wird, weil seine Breite zu gering ausfällt, oder weil der höhere Kanal in den Bereich des niedrigeren hereinragt. Man spricht dann von „missing codes“.



Quelle: <https://de.wikipedia.org/wiki/Analog-Digital-Umsetzer>

Abbildung 13: Stufungsfehler, Breite der Stufe(links), missing codes(rechts)

### 4.3. Realisierung

In diesem Kapitel werden nur kurz die verschiedenen Verfahren zur Realisierung solcher ADCs angegeben. Eine komplette technische Einführung führt an dieser Stelle zu weit.

#### 4.3.1. Komparator-Verfahren

Bei Komparator- oder sogenannten Zählverfahren passieren zwei Dinge gleichzeitig. Auf der einen Seite findet ein analoger Prozess statt der durch die zu messende Größe bedingt wird. Man lädt zum Beispiel einen Kondensator auf, der mit einem Komparator verbunden ist. Daneben findet ein digitaler Prozess statt, welcher Zeiten oder Impulsdichten misst.

Im einfachsten Fall funktioniert es nun so: Man legt einen Sägezahn an einen Komparator und die Eingangsspannung der zu messenden Größe. Es wird dabei die Zeit gemessen, bis die ansteigende Sägezahnspannung genau so groß ist, wie der analoge Eingang am Komparator. Der Steigung des Sägezahn ist konstant. Also ist die gemessene Zeit direkt proportional zu der angelegten Spannung und somit proportional zur Kanalnummer.

#### 4.3.2. Rückgekoppelte Umsetzer

Diese arbeiten seriell mit sogenannten DAC (Digital Analog Converter), welche das genaue Gegenstück zu ADCs sind, sie machen aus einem digitalen Signal ein analoges. Diese Schaltung funktioniert wie folgt: Man läuft bei dem DAC alle möglichen Digitalwerte von unten nach oben durch und vergleicht den Ausgang des DACs mit dem analogen Messsignal. Stimmen diese beiden überein, bzw. wird der Ausgang des DACs zum ersten mal größer als das Eingangssignal, so ist der am DAC eingestellte Digitalwert auch direkt die Kanalnummer des ADCs.

## 4. Analog Digital Wandler

---

### 4.3.3. Parallel Umsetzer

Diese Methode ist deutlich schneller, und funktioniert durch nur einen einzigen Vergleich. ADCs, die dieses Verfahren nutzen werden auch Flash-ADCs genannt. Das Prinzip ist denkbar einfach, jeder Kanal besitzt einen eigenen Komparator. Das Eingangssignal wird an allen Komparatoren gleichzeitig angelegt und alle, die auslösen senden eine binäre 1. Somit ergibt sich die Kanalnummer dann als Binärzahl. Der einzige Nachteil an dieser Methode ist, dass man für ADCs mit großer Bittiefe, extrem viele Komparatoren braucht. Für einen 12-Bit Umwandler, sind dies schon  $2^{12} - 1 = 4095$ .

### 4.4. Wichtige Kenngrößen

In dieser Liste finden sich die wichtigsten Parameter oder Kenngrößen, die man über das ADC in Erfahrung bringen sollte (Datenblatt oder Messungen) oder durch richtiges Messen verhindern kann, sodass sie keine Rolle mehr spielen.

- Abtastrate oder Sampling Rate - Dies gibt die momentane oder maximal mögliche Anzahl an Umwandlungen oder auch Samples pro Sekunde an
- Auflösung oder Resolution - Wie breit eine Stufe bzw. Kanal im ADC ist, also die kleinstmögliche Schritte in welchen das ADC messen kann
- Nullpunktsfehler - Ist meine Null von dem im Datenblatt angegeben Nullpunkt verschoben, wenn ja um wie viel
- Verstärkungsfehler - ist meine Umsetzungskennlinie mit einer falschen Steigung angegeben
- Integrale Nichtlinearität - Ein Fehler der dadurch entsteht, dass meiner Umsetzungskennlinie eine Gerade zugrunde gelegt wird, welche allerdings in der Realität keine ist
- Differentielle Nichtlinearität - Sind die Stufen in meinem ADC unterschiedlich breit zueinander. Manche Kanäle werden bei gleichmäßig verteilter Eingangsspannung unterschiedlich oft getroffen
- Quantisierungskennlinie - Grafische Darstellung zwischen analogem Eingangswert und digitalen Ausgangswert (Umsetzungskennlinie)
- Quantisierungsfehler - Abweichung von Eingangs- und Ausgangswert aufgrund der begrenzten Auflösung.
- Informationslücken oder missing codes - Bei kontinuierlicher Vergrößerung des Eingangssignals werden Kanäle übersprungen. Die ist möglich wenn die Differentielle Nichtlinearität größer als ein LSB (Least Significant Bit) ist.
- Latenzzeit - Verzögerung zwischen Eingang des Signals und umgewandelten Ausgang. Dies ist vor allem bei Steuertechnik interessant.
- Signal-zu-Rauschen Verhältnis oder Signal to Noise Ratio in dB

## 4. Analog Digital Wandler

---

- Dynamischer Bereich - In welchem Bereich erwartet mein ADC Eingänge und welcher Art (zum Beispiel von 0 - 5 Volt, oder 0 - 100 pico Coulomb).

### 4.5. Verwendete Geräte

In dieser Arbeit kamen zwei verschiedene ADCs zum Einsatz. Der eine wurde für die Aufnahme von Einzelpulsen, der andere für hohe Raten benutzt, bei welchem sich die Pulse zu einer Gleichspannung aufaddieren.

#### 4.5.1. ADC CAEN V965

Es handelt sich um ein kommerzielles VME-Modul von CAEN. Es kann über einen Master Controller oder einen PC in einem VME-Crate bedient werden. In diesem Fall wurde es über einen Linux PC gesteuert, und war dort schon in ein Testprogramm zur Datenerfassung eingebunden [m].

Es handelt sich um ein ADC-Board mit 16 Signaleingängen mit einer Eingangsimpedanz von  $50 \Omega$ . Man kann also per Board 16 Signale auf einmal verarbeiten. Bei jedem dieser Eingänge wird die deponierte Ladung über ein gewissen Zeitraum aufgenommen und in eine äquivalente Spannung umgewandelt. An den Eingängen werden nur negative Spannungen oder Ströme erwartet. Das Gerät besitzt zwei dynamische Bereiche. Deshalb wird jedes Signal auch von zwei ADC-Chips bearbeitet. Jedes gibt dann die Kanalnummer für seinen entsprechenden dynamischen Bereich aus.

Die Bereiche werden low und high Range genannt. Für die low Range ist der dynamische Bereich von 0 – 100 pC mit einer Auflösung von 25 fC LSB (Least Significant Bit, also 1 Kanal hat die Breite 25 fC) und für die high Range ist der Bereich 0 – 900 pC mit einer Auflösung von 200 fC LSB.

In dieser Arbeit wurde nur mit den high Range Bereich gearbeitet, da der low Range Bereich fehlerhaft war. Wenn man das Gerät in diesem Bereich benutzen wollte, war das Signal so stark verrauscht, dass man mehr Informationen in dem high Range bereich hatte (auch für kleine Signale), als im low Range.

Die Ausgabe des ADCs erfolgte über den VME-Bus in einem 32 Bit Wort. Um den Zeitraum festzulegen, während der Ladung gesammelt wird, besitzt das Modul einen Eingang, das sogenannte GATE. Dieser Eingang erwartet einen Nim-Signal (also -0.8V Gleichspannung). Solange dieses Signal anliegt, wird gesammelt, wenn es aufhört, umgewandelt. In Abbildung 14 ist das Board, welches benutzt wurde, von der Front zu sehen.

Im Zuge dieser Arbeit wurde dieses Board benutzt, um den Unterschied in der Verstärkung der neuen PMT-Base zu messen. Da es sich hierbei um eine relativ grobe Messung handelt, die den Faktor nur auf etwa 5% genau messen soll, wurde darauf verzichtet alle in Kapitel 4.4 angesprochen Werte zu messen.

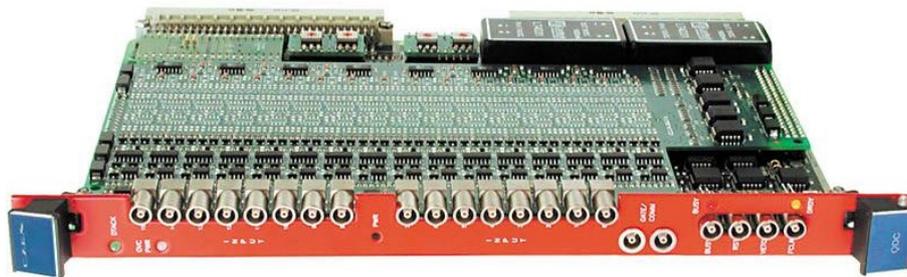


Abbildung 14: Foto der Front des CAEN Boards

##### 4.5.2. VME QWEAK Modul von TRIUMF

Hierbei handelt es sich um ein speziell für das QWEAK-Experiment entwickeltes ADC Board. Das QWEAK-Experiment nutzt ebenso wie P2 die Messung der paritätsverletzenden Asymmetrie im Wirkungsquerschnitt der elastischen Elektron-Proton-Streuung, um die schwache Ladung des Protons bei kleinem Impulsbetrag zu bestimmen.

Freundlicherweise wurde uns zum Testen eines der verbleibenden Module zur Verfügung gestellt. Das Gerät ist ein 18-Bit ADC mit 8 Signaleingängen und einer Eingangsimpedanz von  $12\text{ k}\Omega$ . Zusätzlich besitzt es noch zwei Eingänge: Das sogenannte Gate und die Clock. Mit dem Gate gibt man bei diesem Gerät an, ab wann eine Konversionsperiode anfängt. Die Clock gibt den Takt, an mit dem das Gerät arbeitet. Das bedeutet, dass auch die maximale Abtastrate von der Clock abhängig ist. Standardmäßig gibt es einen inneren Takt von 20MHz, was auch der maximal mögliche Takt für das Gerät ist.

Die maximale Abtastrate (Sampling Rate) beträgt  $500\text{ kSP/s}$  (kilo Samples pro Sekunde). Es kann also maximal alle zwei Mikrosekunden ein Wert abgetastet werden. Das Gerät hat einen dynamischen Bereich von  $\pm 10\text{V}$  mit einer Auflösung von  $76.29\text{ }\mu\text{V LSB}$  (Least Significant Bit, bedeutet in diesem Fall, eine Kanalnummer hat die Breite von  $76.29\text{ }\mu\text{V}$ ). Der Nullpunkt liegt nicht genau bei Kanalnummer 0 sondern bei -375, was einer Abweichung von  $-28.61\text{ mV}$  entspricht. Da für alle weiteren Messungen immer sicherheitshalber eine Messung der Null durchgeführt wurde, ist dies für die weitere Arbeit nicht von Belang.

Das Board hat eine weitere Besonderheit: Aufgrund eines FPGAs, welches auf dem Board angebracht wurde, kann man bei diesem Board steuern, wieviele Samples nach Einsetzen des Gate-Pulses aufgenommen werden. Das Board kann maximal 65532 Samples aufnehmen, bevor es an die Umwandlung geht. Dies entspricht bei der schnellsten Sampling Rate einem Zeitraum von 131.064 ms. Die jeweils ausgelesene Kanalnummer wird aufaddiert, um so eine höhere Auflösung zu erzielen wie mit Einzelmessungen möglich. Am Ende der Sampleperiode gibt der ADC nun nur eine Kanalnummer zurück, die deutlich größer ist, als eigentlich bei 18-Bit möglich sind.

Der verwendete ADC-Chip ist ein AD7674 von Analog Devices. Wie auch das gesamte Board hat er eine 18-Bit Auflösung ohne Informationslücken. Laut Hersteller ist die Integrale Nichtlinearität  $\pm 2.5\text{ LSB}$  und das Signal-zu-Rauschen Verhältnis  $100\text{ dB}$ . In Abbildung 15 findet sich ein Foto des ADCs.

Da dieser ADC neu aus Amerika geschickt wurde, musste es erst einmal in die bestehende Datenerfassung eingebunden werden. Dafür mussten Teile der Software, die beim QWEAK-Experiment

#### 4. Analog Digital Wandler

---

verwendet wurden und für ein VxWorks System geschrieben wurden in C und C++ umgewandelt und in die bestehende P2 Datenaufnahme integriert werden. Bei VxWorks handelt es sich um ein spezielles Betriebssystem.

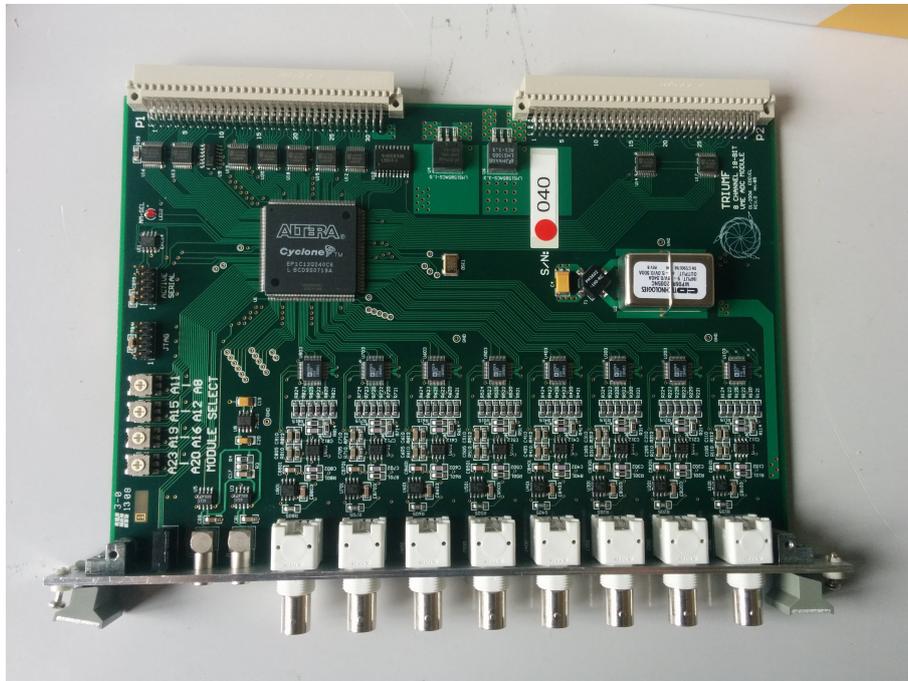


Abbildung 15: Foto des des QWEAK Boards

Im nächsten Abschnitt finden sich Messungen zur Linearität und dem Rauschverhalten dieses ADCs und zum Abschluss eine Messung aus einer Teststrahlzeit an MAMI.

### 5. Messung und Auswertung

Hier werden die verschiedenen Messungen, die mit dem umschaltbaren Spannungsteiler und dem TRIUMF-ADC durchgeführt wurden, genauer erklärt und dargestellt.

#### 5.1. Verstärkungsänderung

Um die besondere Verstärkungsschaltung des Spannungsteilers zu testen, wurde ein Testaufbau verwendet, der in Abbildung 16 skizziert ist.

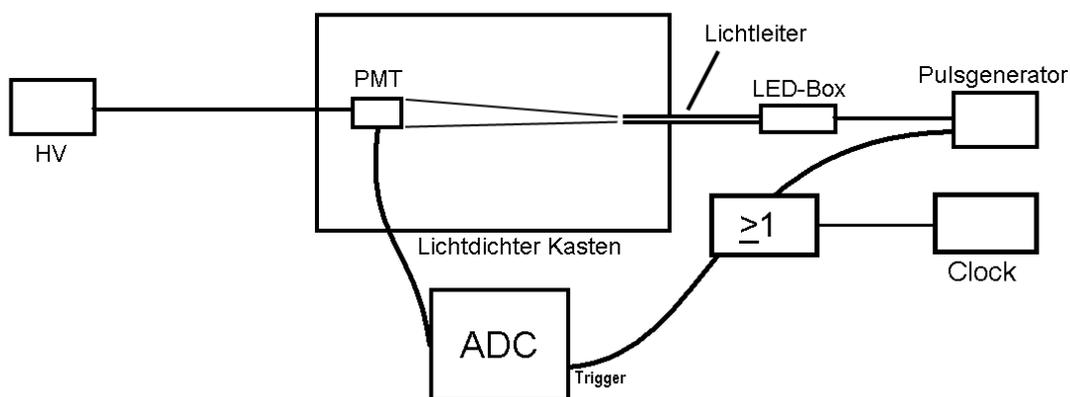


Abbildung 16: Schema zum Testen der relativen Verstärkung

Eine blaue LED mit Wellenlänge 465 nm die mit einem Lichtleiter in einen lichtdichten Kasten geführt wird, bescheint den PMT. Die LED wird mit einem Pulsgenerator betrieben, beleuchtet den PMT also nur für kurze Zeit in bestimmten Intervallen. In diesem Fall wurde eine Frequenz von 1 kHz gewählt und eine Pulsbreite von 80 ns bei einer Pulstiefe von etwa 1.8 Volt. Das Signal des PMTs führt zum ADC. Durch den Pulsgenerator gibt es nur ein bestimmtes Fenster, in dem Pulse des PMTs auch durch die LED ausgelöst werden. Eine Koinzidenzeinheit verschaltet das Triggersignal des Pulsers mit einer Clock durch ein ODER-Gatter. Dieses Endsignal ist mit dem ADC-GATE verbunden. Der ADC nimmt also immer dann das Signal aus dem PMT auf, wenn wenn der Pulsgenerator die LED ausgelöst hat oder aber wenn nur das Untergrundrauschen als Signal anliegt.

Bei diesen Einstellungen emittiert die LED eine stabile Anzahl Photonen, die nun detektiert werden können. Dies wird bei einer festen Betriebsspannung des PMTs durchgeführt. Danach schaltet man die Relais ein und wiederholt die Messung. Bei dem ADC handelt es sich um das Ladungssensitive integrierendes Modul von CAEN (V965). Da die Verstärkung des PMTs durch das Einschalten der Relais abnimmt, ist beim einstellen des Pulsgenerators darauf zu achten, dass das Peak welches man aufnehmen will, noch nicht im Rauschen verschwindet. Man muss sowohl bei aus- als auch bei eingeschalteten Relais die gleichen Einstellungen benutzen, nur so bleibt das Ergebnis vergleichbar.

## 5. Messung und Auswertung

Wie schon im Kapitel 3.5.1 erwähnt, kann die Verstärkung des Spannungsteilers umgeschaltet werden. Also nimmt man jedes Spektrum einmal mit aus- und einmal mit eingeschalteten Relais auf. Da sich dabei Spannung zwischen den Dynoden ändert, muss man die Spannung für eingeschaltete Relais um den Faktor:  $U_{ein} = \frac{0,225}{0,275} \cdot U_{aus}$  korrigieren. Das bedeutet es gibt zu jeder Betriebsspannung bei ausgeschalteten Relais eine korrespondierende bei eingeschalteten Relais. Die Paare sind für die hier durchgeführten Messungen sind in nachfolgender Tabelle :

Relais aus	Relais ein
1500 V	1227 V
1400 V	1145 V
1300 V	1064 V
1200 V	982 V

Für den weiteren Verlauf und zum einfacheren Verständnis wird für ausgeschaltete Relais öfter der Ausdruck High Gain Mode und für eingeschaltete Relais öfter der Ausdruck Low Gain Mode verwendet.

In den Abbildungen 17 und 18 sieht man das Ergebnis einer solchen Messung. Auf jedem der Spektren sind 2 Peaks zu sehen. Bei dem Peak der etwa bei Kanal 69 zu sehen ist, handelt es sich um das sogenannte Pedestal-Peak. Es kommt von dem Triggersignal der Clock, nimmt also nur das Hintergrundrauschen auf. Damit korrigiert man einerseits den Offset der Null des ADCs, und zum anderen kann es sein, dass auf dem Eingang durch die Elektronik des PMTs eine geringe Offsetspannung anliegt, auch wenn eigentlich kein Signal zu sehen ist. Beide Offsets können durch die Messung des Pedestals korrigiert werden. Man sieht auch an der Breite des Pedestals, ob das Untergrund-Signal verrauscht ist. Dann wäre der Peak sehr viel Breiter als 2-3 Kanäle

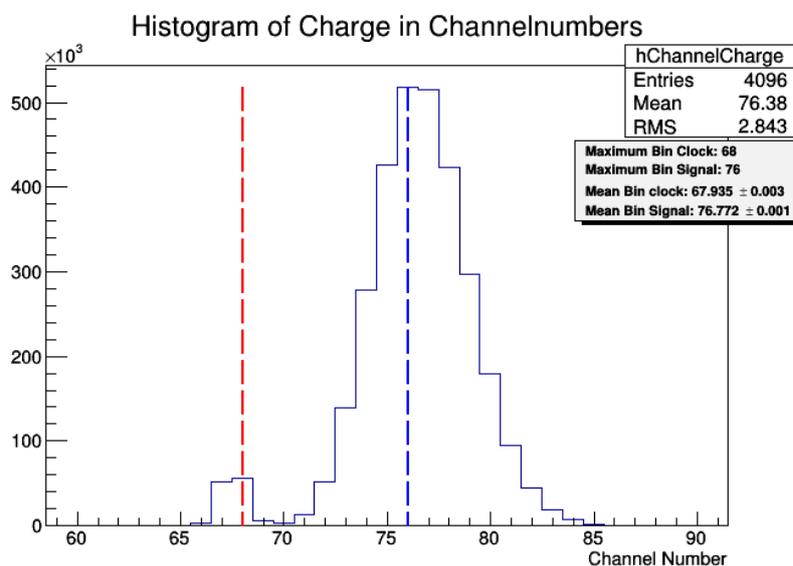


Abbildung 17: 1200V  $U_{Versorgung}$  Relais aus (High Gain Mode)

In den Spektren wurde jeweils der Bin mit den meisten Einträgen mit einem Strich markiert. Die Clock mit rot und das Signal in blau. Im breiteren der beiden Peaks, also bei Kanal 261 im High

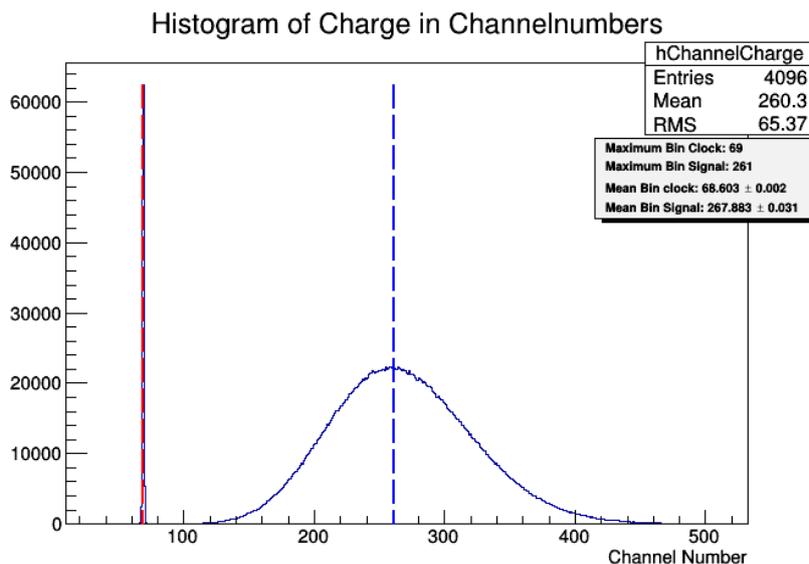


Abbildung 18: 982V  $U_{V\text{ersorgung}}$  Relais ein (Low Gain Mode)

Gain Mode bzw. Kanal 77 im Low Gain Mode befinden sich die Ereignisse, die von den durch den LED-Pulser ausgelösten PMT-Signalen stammen. Um die Verstärkung zu berechnen, zieht man den ADC-Kanal des Clock-Peaks von dem des Signal-Peaks ab. Die Lage der Peaks wurde auf zwei Arten bestimmt. Einmal wurde einfach der Bin mit den meisten Einträgen genommen und einmal wurde nach der Wahl passender Ober- und Untergrenzen jeweils der arithmetische Mittelwert ausgerechnet. Daraus ergeben sich mitunter größere Unterschiede je nach Form der Spektren.

Für symmetrische Peaks wäre der Unterschied sehr klein bzw. nicht vorhanden, da bei realen Messungen jedoch kein Peak wirklich symmetrisch ist, ergeben sich wie in den folgenden Tabellen mehr oder weniger große Unterschiede. Man kann dies direkt auch in Abbildung 18 sehen, wo man augenscheinlich die Mitte des Spektrums ein Bin weiter rechts gesetzt hätte. Da hier das Signal und die gemessene Null nur etwa 8 Bins auseinander liegen, ist dies ein Unterschied von 12.5%.

Methode	$U_V$ Relais aus	Clock Bin	Signal Bin	Signal Korrigiert
hoechster Bineintrag	1500 V	69	1174	1105
hoechster Bineintrag	1400 V	69	743	674
hoechster Bineintrag	1300 V	69	444	375
hoechster Bineintrag	1200 V	69	261	192
arith. Mittelwert	1500 V	$68.603 \pm 0.002$	$1246.373 \pm 0.176$	$1177.770 \pm 0.176$
arith. Mittelwert	1400 V	$68.624 \pm 0.002$	$773.182 \pm 0.105$	$704.558 \pm 0.105$
arith. Mittelwert	1300 V	$68.602 \pm 0.002$	$455.428 \pm 0.059$	$386.826 \pm 0.059$
arith. Mittelwert	1200 V	$68.603 \pm 0.002$	$267.883 \pm 0.031$	$199.280 \pm 0.031$

## 5. Messung und Auswertung

Methode	$U_V$ Relais ein	Clock Bin	Signal Bin	Signal Korrigiert
hoechster Bineintrag	1227 V	68	100	32
hoechster Bineintrag	1145 V	67	88	21
hoechster Bineintrag	1064 V	67	81	14
hoechster Bineintrag	982 V	68	76	8
arith. Mittelwert	1227 V	$67.540 \pm 0.002$	$100.522 \pm 0.005$	$32.982 \pm 0.005$
arith. Mittelwert	1145 V	$67.437 \pm 0.002$	$88.288 \pm 0.003$	$20.851 \pm 0.004$
arith. Mittelwert	1064 V	$67.521 \pm 0.002$	$81.087 \pm 0.002$	$13.566 \pm 0.003$
arith. Mittelwert	982 V	$67.935 \pm 0.003$	$76.772 \pm 0.001$	$8.837 \pm 0.003$

Bevor nun aber die Änderung in der Verstärkung gemessen wird, kann, wie in Kapitel 3.3 beschrieben, überprüft werden, ob sich dieser spezielle Spannungsteiler bei der Messung der Verstärkung in Abhängigkeit der Betriebsspannung wie erwartet verhält. Man vergleicht nun jeweils die vier Messungen im High und Low Gain Mode untereinander. Verhält sich der Spannungsteiler so wie erwartet, müsste sich bei doppelt logarithmischen Darstellung ein linearer Zusammenhang ergeben. Es wurde an dieser Stelle nur mit den arithmetischen Mittelwerten gearbeitet, da dies die genauere Methode zur Bestimmung der Verstärkung darstellt. Außerdem kann hier die Unsicherheit des Mittelwerts mithilfe der Formel zur Standardabweichung abgeschätzt werden. In Abbildung 19(a) sieht man den Verlauf für ausgeschaltete Relais (High Gain). Daneben sieht man in 19(b) den sogenannte Residuen-Plot für diese Daten. Bei diesem wird die Differenz aus Datenpunkten und Fitwert gebildet. Die Graphen wurden mit ORIGIN gefittet.

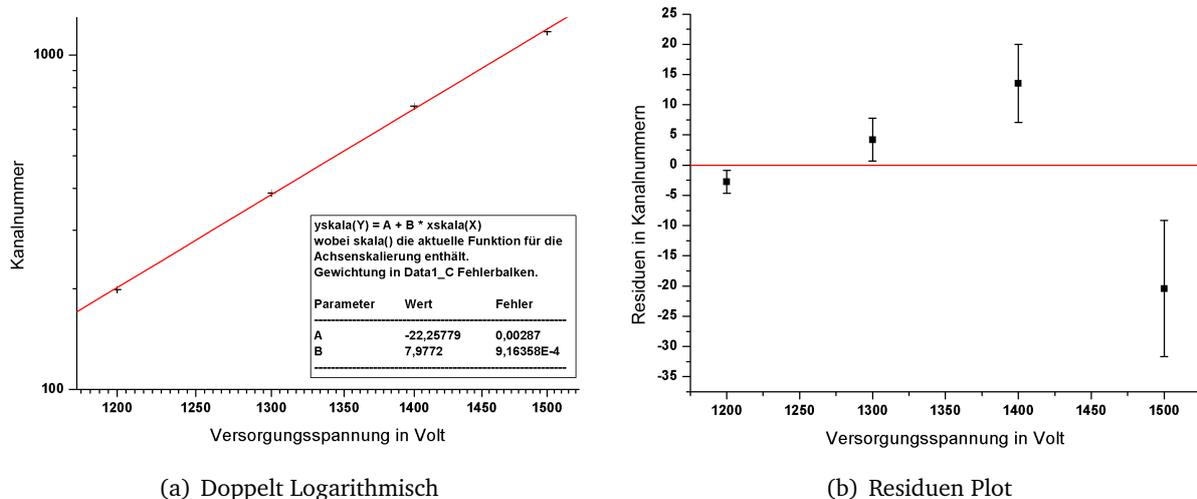


Abbildung 19: Verstärkung im High Gain Mode

Wie man sieht, bestätigt sich bei doppeltlogarithmischer Auftragung der lineare Zusammenhang zwischen Betriebsspannung und Signalwert des PMTs. Messwerte und Geradenfit weichen weniger als drei Standardabweichungen voneinander ab. Es sieht allerdings so aus, als würden die Messwerte für größere Versorgungsspannungen immer weiter von der Gerade abweichen. Führt man einen Fit nur für die ersten drei Messwerte durch und schaut sich dort den Residuen Plot an, so sieht man in Abbildung 20, dass die Werte deutlich besser mit einem linearen Fit übereinstimmen.

## 5. Messung und Auswertung

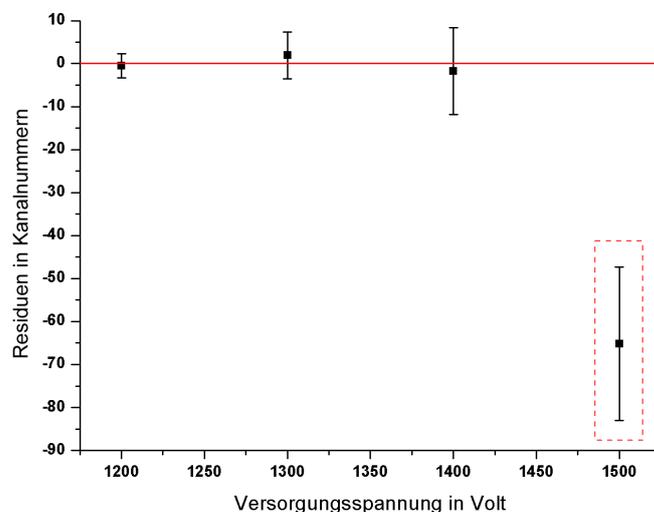


Abbildung 20: Verstärkung im High Gain Mode, Residuen Plot

Der Wert für 1500 Volt wurde wie oben geschrieben nicht im Fit betrachtet. Er wurde deshalb gesondert markiert.

Für höhere Versorgungsspannungen steigt die Verstärkung des PMTs also schwächer an als theoretisch vermutet. Dies kann mehrere Gründe haben. Man befindet sich an der oberen Grenze für die Verstärkung des PMTs, oder die Anzahl an Elektronen wird bei dieser hohen Betriebsspannung in den letzten Stufen so groß, dass auch der aktiv geregelte Spannungsteiler nicht spannungsfest an den einzelnen Dynoden bleibt.

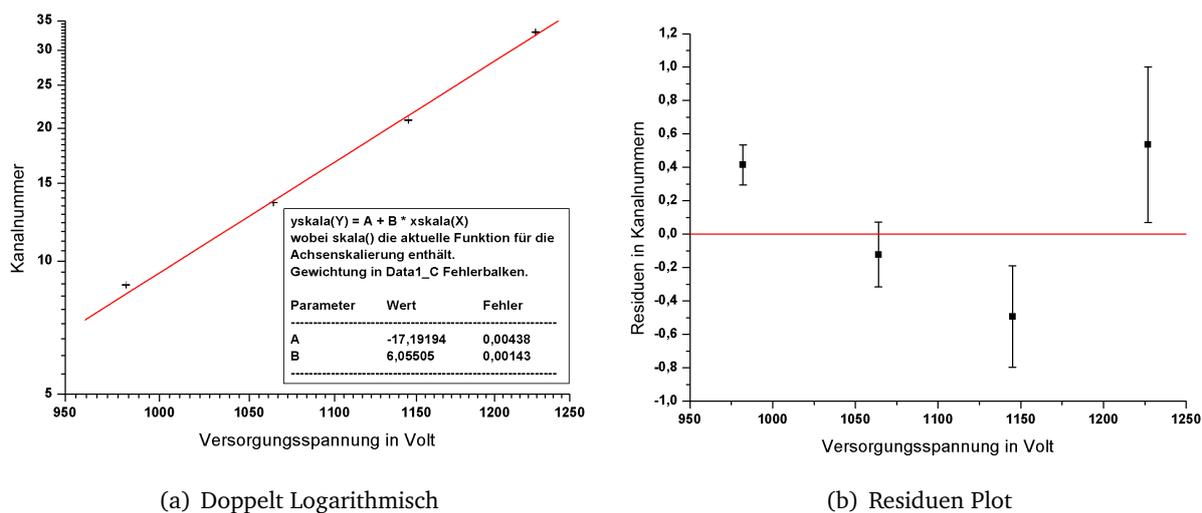


Abbildung 21: Verstärkung im Low Gain Mode

Die gleiche Betrachtung wird nun für eingeschaltete Relais (Low Gain Mode) wiederholt. Dies

## 5. Messung und Auswertung

---

ist wichtig, da man an dieser Stelle durch das Überbrücken von Dynoden in die Sammeloptik des PMTs eingegriffen hat. In Abbildung 21(a) und 21(b) kann man das Ergebnis sehen.

Als erstes sieht man auch hier, dass der Verlauf nicht signifikant verschieden von der linearen theoretischen Voraussage ist. Auch hier scheint bei höheren Betriebsspannungen die Abweichung zum Fit größer zu werden, allerdings in genau umgekehrter Manier als im High Gain Mode. Bei höheren Betriebsspannungen ist die Verstärkung höher als vom Fit vorausgesagt. Der Grund dafür kann an dieser Stelle nicht geklärt werden.

Es ist jedoch sehr auffällig, dass jeweils die Messung für die höchste Betriebsspannung so deutlich neben dem Fitverlauf liegt. Die erste Vermutung wäre, dass sich zum Beispiel die Einstellung des Funktionsgenerators am LED-Pulsers geändert haben. Dagegen spricht allerdings, dass die Abweichung dann in die gleiche Richtung hätte gehen müssen. Dies gilt für alle systematisch eingeführten Fehler. Der Fakt, dass es im High Gain zu niedrig und im Low Gain zu hoch gemessen wurde, macht eine einfache Fehlmessung unwahrscheinlich.

Auch hier wurde noch einmal ein Residuen-Plot erstellt, der nur die ersten drei Punkte mit in den Fit einbezieht, zu sehen in Abbildung 22.

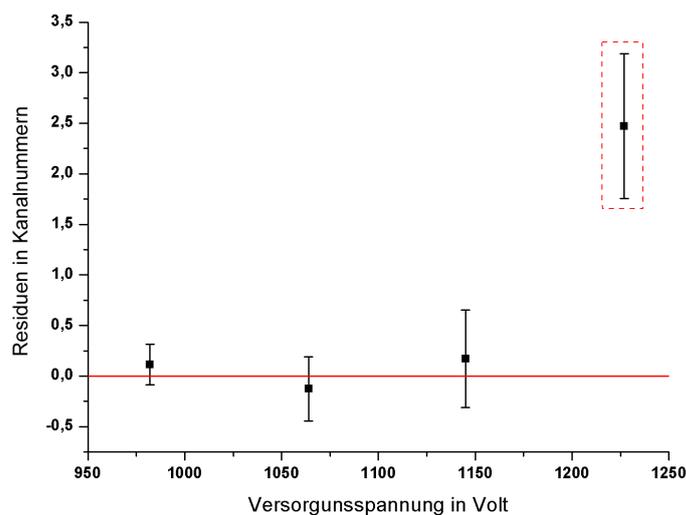


Abbildung 22: Verstärkung im Low Gain Mode, Residuen Plot

Wie man sieht, stimmen so die ersten drei Punkte deutlich besser überein, als wenn man alle vier Punkte mit einbezieht.

Dafür ist der vierte Punkt innerhalb der Fehlerbetrachtung nun weiter als drei Standardabweichungen von der Fitgeraden entfernt. Nach Betrachtung des Low Gain Modes, kann ausgeschlossen werden, dass die Verstärkung aufgrund der Limitierung des PMTs zu schwach gestiegen ist im High Gain Modus. Wäre dies der Fall, so hätte der Wert jetzt, nachdem der PMT effektiv um zwei Verstärkungsstufen gekürzt wurde zum restlichen linearen Zusammenhang passen müssen oder ebenfalls als zu klein gemessen worden sein.

Da nun festgestellt wurde, dass der PMT trotz des speziellen Spannungsteilers in beiden Modi zumindest für die kleineren Betriebsspannungen so funktioniert wie vorausgesagt, kann man

## 5. Messung und Auswertung

---

nun die gemessenen Werte für die jeweiligen Paare durcheinander teilen, und so die Änderung in der Verstärkung bestimmen. In der unten stehenden Tabelle sind die Ergebnisse eingetragen.

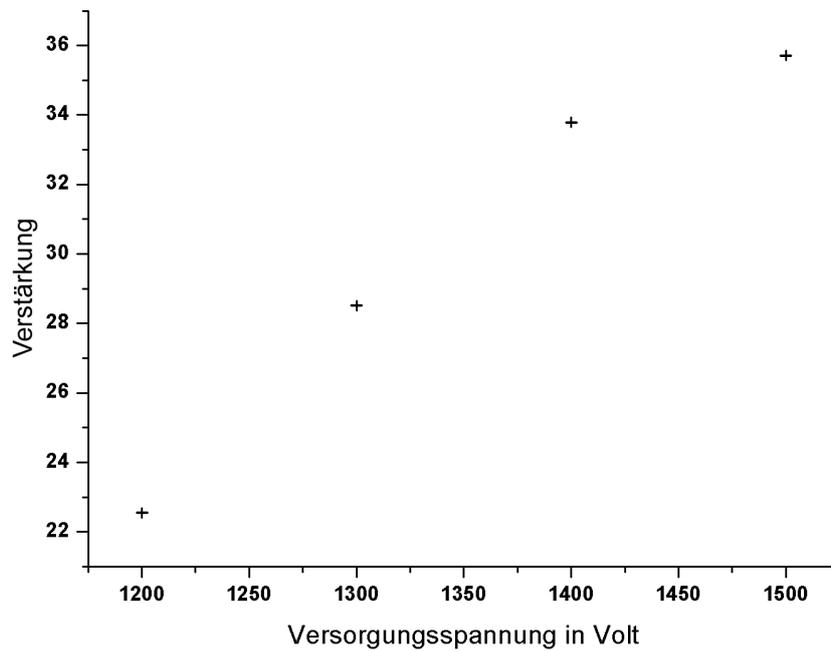
$U_V$ Relais aus	Verst. Faktor Beide		Verst. Faktor Einzel	
	hoechster Bineintrag	arith. Mittelwert	hoechster Bineintrag	arith. Mittelwert
1500 V	34.531	$35.709 \pm 0.008$	5.876	$5.976 \pm 0.000$
1400 V	32.095	$33.790 \pm 0.008$	5.665	$5.813 \pm 0.000$
1300 V	26.786	$28.514 \pm 0.007$	5.175	$5.340 \pm 0.000$
1200 V	24.000	$22.551 \pm 0.009$	4.899	$4.749 \pm 0.000$

Bei der Methode „höchster Bineintrag“ wurden keine Fehler mit angegeben, da an dieser Stelle lediglich der Bin mit den meisten Einträgen abgelsen wurde. Man sieht, dass für kleinere Versorgungsspannungen die relative Verstärkung absinkt, was auch nicht weiter verwunderlich ist, da die Verstärkung jeder einzelnen Dynode abnimmt und mit der speziellen Schaltung einfach zwei von diesen ueberbrückt werden.

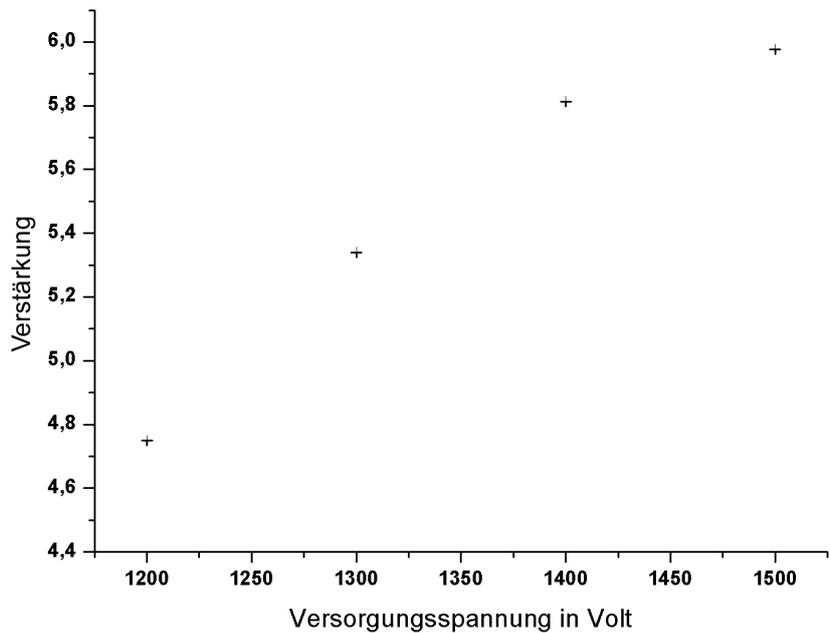
Deshalb ist auch die geringe Änderung ( $< 100$ ) in der Verstärkung nicht weiter verwunderlich, da laut Hamamatsu (Vermerk in die Quellen) und Philips (Vermerk in die Quellen) die Verstärkung einer herkömmlichen Dynode bei etwa 4-6 liegt. Fallen zwei davon weg, verringert sich die Verstärkung also um Faktor 16-36. Will man eine noch höhere Änderung erreichen, muss man weitere Dynoden ueberbrücken oder die Hochspannung ändern (bzw. eine Kombination aus beiden). Bei den gemessenen Werten bewegt man sich zwischen einer Verstärkungsänderung von 22.6 und 35.7. Um die Verstärkung einer einzelnen Dynode zu erhalten, muss nun nur die Wurzel dieses Wertes gezogen wird. Daraus folgt eine Verstärkung pro Dynode von etwa 4.8 bis 6.0 je nach Versorgungsspannung. Würde man nun noch eine weitere Dynode ueberbrücken, so ergäbe sich rein rechnerisch eine Verstärkungsänderung von etwa 108-214. Dies entspricht eher der Größenordnung, in welcher sich die gewünschte Änderung bewegen sollte und sollte in späteren Prototypen getestet werden.

Das ueberbrücken von Dynoden ist allerdings nicht beliebig möglich, da man aufgrund der komplizierteren Schaltung die Pulse verformt (durch die immer längeren Wege über immer mehr Relais werden Kapazitäten und Induktivitäten zur bestehenden Schaltung hinzugefügt). Dies stellte allerdings für diese wenigen Dynoden noch kein Problem dar.

Der Hersteller gibt für die verwendete Röhre als typische Betriebsspannung einen Wert von 1000 Volt an. Maximal kann man den PMT allerdings mit Spannung bis zu 2700 Volt betreiben. Bei den Messungen wurde sich also deutlich unterhalb der maximal zulässigen Spannung bewegt. Als letztes wurde noch die Änderung in dem Verstärkungsfaktor über die verschiedenen Betriebsspannungen aufgetragen. Einmal für beide Dynoden und einmal für die Änderung pro Dynode. Als Betriebsspannung wurde hier die Spannung im High Gain Mode eingetragen.



(a) Beide Dynoden



(b) Einzelne Dynode

Abbildung 23: Verstärkungsänderung des Spannungsteilers

Man sieht für höhere Betriebsspannungen ein Ansteigen der Verstärkung. Da die gesamte Verstärkung mit steigender Hochspannung ansteigt und somit auch die Verstärkung jeder einzelnen

## 5. Messung und Auswertung

---

Dynode, ist das sinnvoll. Man sieht auch, dass die Steigung nicht konstant bleibt sondern immer geringer wird. Das lässt sich natürlich für den letzten Wert direkt auf den zu hohen Low Gain Mode Wert und den zu niedrigen High Gain Mode Wert zurückführen.

Ein funktionaler Zusammenhang müsste sich aus den Fits für die Einzelmessungen ergeben.

Die generelle Fitfunktion die bei einem doppeltlogarithmischen Plot einen linearen Fit ergibt sieht wie folgt aus:

$$f(u) = 10^{a+b \cdot \lg(u-c)} \quad (15)$$

Die Parameter sind aus den vorherigen Fits bekannt und befinden sich in der folgenden Tabelle.

Bereich	Parameter		
	a	b	c
High Gain	$-22.25779 \pm 0.00287$	$7.97720 \pm 0.00091$	1
Low Gain	$-17.19194 \pm 0.00438$	$6.05505 \pm 0.00143$	0.225/0.275

Der Parameter  $c$  ist für die Korrektur der Spannung des Low Gain Modes um am gleichen Arbeitspunkt zu bleiben.

Eine Division der beiden Gleichungen ergibt wie auch schon bei den Messwerten die Funktion der Verstärkungsänderung.

$$f(u) = 2.895 \cdot 10^{-5} \cdot u^{1.92} \quad (16)$$

Diese Funktion wird nun zusätzlich in 23(a) eingetragen. Das Ergebnis ist in 24 in rot zu sehen. Die berechnete Funktion widerspricht den Daten eindeutig. Dies ist nicht weiter verwunderlich, da wie schon in 19(b) und 21(b) gesehen, die Messwerte für die höchsten Betriebsspannung nicht gut mit der Theorie übereinstimmen

Betrachtet man auch hier wieder nur die ersten 3 Messungen, so ergeben sich die Fitparameter zu:

Bereich	Parameter		
	a	b	c
High Gain	$-22.92481 \pm 0.00435$	$8.19223 \pm 0.001410$	1
Low Gain	$-15.87954 \pm 0.00749$	$5.62151 \pm 0.002400$	0.225/0.275

Geht man hier analog vor, so ergibt die folgende Gleichung

$$f(u) = 8.24665 \cdot 10^{-7} \cdot u^{2.57} \quad (17)$$

Diese Kurve ist auch in 24 in grün zu sehen.

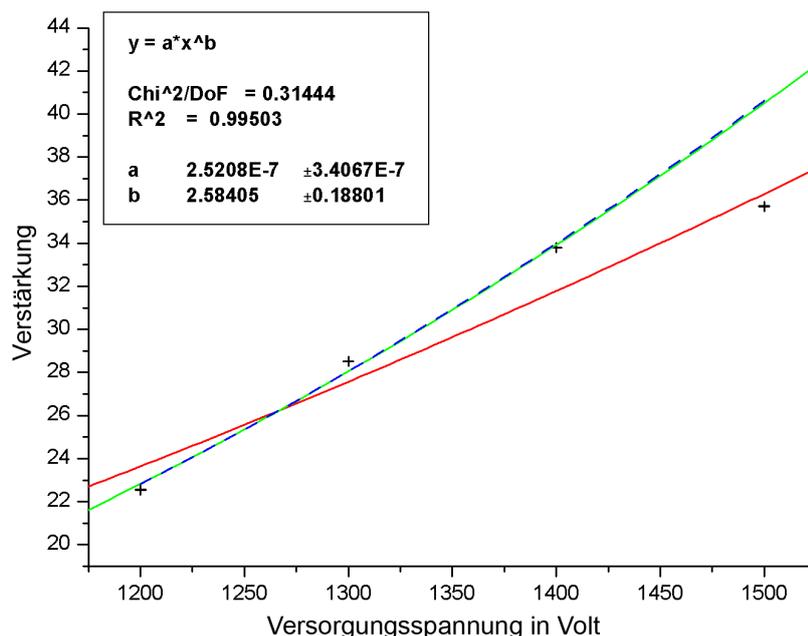


Abbildung 24: Verstärkungsänderung beider Dynoden in Abhängigkeit der Betriebsspannung

Als letztes wurde nun ein Fit anhand der Funktionsvorschrift:  $a \cdot x^b$  durchgeführt da auch die errechneten Gleichungen eine ähnliche Form haben. Dabei sind  $a$  und  $b$  Parameter,  $x$  die Variable. Dies dient dazu zu sehen, ob die mit den Fitgeraden berechnete Gleichung auf ein ähnliches Ergebnis kommt wie ein direkter Fit. Da Fitten ein komplexer Prozess ist, ist er nicht intuitiv distributiv. Das Ergebnis ist im selben Plot in Blau gestrichelt zu sehen.

Die daraus resultierende Funktion lautet:

$$f(u) = 2.521 \cdot 10^{-7} \cdot u^{2.584} \quad (18)$$

Wie man im Plot sieht, sind die blaue und die grüne Kurve nahezu identisch. Dies sieht man auch an der Funktionsvorschrift, welche nur einen geringen Unterschied in den Parametern haben. Innerhalb der Fehlerbetrachtung sind die beiden Gleichungen sogar identisch.

Es ist noch zu beachten, dass die theoretisch bestimmte Kurve, da der Exponent  $> 1$  ist, immer steiler wird. Das heisst, dass die Verstärkung immer schneller ansteigen müsste. Dies ist natürlich in der Realität nicht der Fall, da der PMT bei einer gewissen Spannung seine Leistungsgrenze erreicht und durch eine zu hohe Betriebsspannung zerstört wird.

Nachdem alle Aspekte eingängig betrachtet wurden, kann der Grund dafür, dass die letzte Messung nicht zu den restlichen Daten passt immer noch nicht geklärt werden.

In Kapitel 6 wird untersucht ob sich die auf diese Art gemessenen Änderungen in der Verstärkung bei einer Teststrahlzeit am Elektronenbeschleuniger MAMI reproduzieren lassen.

## 5.2. Rauschverhalten des TRIUMF ADCs

Um festzustellen, ob der ADC den in Kapitel 2 genannten Anforderungen genügt, folgt in diesem Unterkapitel eine grobe Abschätzung der Größenordnung in der Unsicherheit der Asymmetrie, welcher allein aufgrund des Eigenrauschens des ADCs auftritt. Man kann damit nur die Größe des Fehlers auf die Asymmetrie abschätzen, nicht die Asymmetrie selbst. Aus Kapitel 2 ist die Formel:

$$A_{gem} = \frac{N_{\uparrow} - N_{\downarrow}}{N_{\uparrow} + N_{\downarrow}} \quad (19)$$

bekannt. Schaut man sich die Fehlerfortpflanzung an, so ist diese durch

$$\Delta A_{gem} = \sqrt{\left(\frac{2N_{\downarrow}}{(N_{\uparrow} + N_{\downarrow})^2} \cdot \Delta N_{\uparrow}\right)^2 + \left(\frac{2N_{\uparrow}}{(N_{\uparrow} + N_{\downarrow})^2} \cdot \Delta N_{\downarrow}\right)^2} \quad (20)$$

gegeben.

Da bei den Tests kein Polarisierter Strahl zur Verfügung steht, also keine verschiedenen Helizitäten, muss man an dieser Stelle einige Annäherungen machen, um den Fehler der Asymmetrie abzuschätzen. Es wird angenommen, dass die Zählraten und deren Fehler gleich hoch sind, also  $N_{\uparrow} = N_{\downarrow} = \frac{1}{2} \cdot N$  und  $\Delta N_{\uparrow} = \Delta N_{\downarrow} = \Delta N$ . Damit vereinfacht sich die obere Formel zu dem kompakten Ausdruck:

$$\Delta A_{gem} = \sqrt{2 \cdot \left(\frac{\Delta N}{N}\right)^2} = \sqrt{2} \cdot \frac{\Delta N}{N} \quad (21)$$

Jetzt misst man bei einem ADC aber keine Zählraten sondern Spannungen, welche durch eine Kanalnummer repräsentiert wird. Wie in Kapitel 4.5.2 beschrieben, kann mit diesem speziellen ADC die Spannung über eine gewisse Zeit immer wieder abgetastet werden, um so eine noch genauere Abstufung innerhalb des dynamischen Bereichs des ADCs zu erlangen.

Aus der Umrechnung von Kanalnummer zu Spannung ist nun die anliegende Spannung während des Tests bekannt. Da man den Eingangswiderstand des ADCs kennt, ist durch die abgefallene gemessene Spannung bekannt, welcher Strom in dem entsprechenden ADC-Eingang fließt. Ein Strom ist nichts anderes als Ladung pro Sekunde. Nun kann man anhand der Elementarladung daraus einen Teilchenstrom machen, und hat somit die gewünschte Größe, Zählrate. Es muss nun nur noch  $\Delta N$  bestimmt werden. Alle Umrechnungen von  $N$  auf  $\mu$  sind in diesem Fall Konstanten, was bedeutet, dass  $\Delta N$  nur von  $\Delta \mu$  abhängt.

Da das Rauschen des ADCs ein statistisches Rauschen sein sollte, erwartet man für die gemessene Kanalnummer eine Gaußverteilung um den wahren Wert  $\mu$ . Da die Messwerte also symmetrisch sein sollten, kann man für  $\mu$  die Formel für das arithmetische Mittel benutzen.

$$\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^n X_i \quad (22)$$

## 5. Messung und Auswertung

---

Es kann ebenfalls eine einfache Formel zur Schätzung der Standardabweichung verwendet werden, sodass man auf einen Fit der gemessenen Kurven verzichten kann, und die beiden Werte direkt aus den Messdaten bestimmt.

$$\sigma = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (X_i - \bar{X})^2} \quad (23)$$

Die Unsicherheit des Mittelwerts  $\Delta\mu$  ergibt sich bei solchen Zählexperimenten zu  $\Delta\mu = \frac{\sigma}{\sqrt{M}}$ , wobei  $M$  die Anzahl der Messpunkte in der Gaußverteilung ist. Setzt man dies nun in die oben vereinfachte Formel für den Fehler der Asymmetrie ein, ergibt sich:

$$\Delta A_{gem} = \sqrt{2} \cdot \frac{\Delta N}{N} = \sqrt{2} \cdot \frac{\sigma \cdot ADC_{resolution}}{Z \cdot e^- \cdot \sqrt{M}} \cdot \frac{Z \cdot e^-}{\mu \cdot ADC_{resolution}} \quad (24)$$

Hierbei ist  $Z$  die Eingangsimpedanz und  $ADC_{resolution}$  die Umrechnung von Kanalnummer zu Volt des ADCs. Wie man sieht, kürzen sich alle Umrechnungsfaktoren und man erhält die übersichtliche Formel:

$$\Delta A_{gem} = \sqrt{2} \cdot \frac{\Delta N}{N} = \sqrt{2} \cdot \frac{\sigma}{\mu \cdot \sqrt{M}} \quad (25)$$

Wie schon in Kapitel 2 geschrieben, ist das Ziel des Experiments ein  $\Delta A^{exp}$  von kleiner als  $4.4 \cdot 10^{-10}$  nach einer Messzeit von 10000 h. Daher muss der Beitrag einzelner Elektronik Bauteile kleiner sein als dieser Wert, oder sehr gut bekannt, damit man überlagerten Gaußkurven später entfalten kann.

Um nun das Eigenrauschen aufzunehmen, wurde der Eingang des ADCs kurzgeschlossen. Dann wurde mithilfe der internen Clock mit einer Frequenz von zehn Kilohertz getriggert. Dies wäre auch die maximale angedachte Frequenz mit der im späteren Experiment gearbeitet werden kann. Wahrscheinlicher sind zur Zeit der Abgabe dieser Arbeit allerdings eher ein bis vier Kilohertz. Mit dieser Triggerfrequenz ergibt sich ein Fenster von 100  $\mu s$ . Da der ADC maximal alle 2  $\mu s$  ein Sample nehmen kann, und etwas Abstand am Anfang und am Ende des Fensters gehalten werden sollte, wurde sich für ein Aufaddieren von 40 Samples entschieden. Damit nutzt man 80 der 100 Mikrosekunden aus. Nach jeweils 80  $\mu s$  gibt der ADC also eine Kanalnummer zurück, welche dann in der Software in einem Array gespeichert wird. Am Ende der Messzeit schreibt die Software die Ergebnisse in ein Textfile. Aufgrund der Art, wie das Programm die Messwerte speichert, gibt es allerdings keine negativen Ausgaben in der Kanalnummer. Ein Beispiel: Die Kanalnummer reichen von  $-10000$  bis  $10000$  für den gesamten dynamischen Bereich von  $-10$  V bis  $10$  V, so gibt das Programm Kanalnummern von  $0$  bis  $20000$  zurück. Dies ändert nichts an den Messergebnissen an sich, man muss nur wissen, dass die Kanalnummern, wenn nicht anders beschrieben, auf einen Bezugspunkt betrachtet werden müssen. Dieser Bezugspunkt ist in allen Fällen der gemessene Nullpunkt oder Untergrund. Deshalb ist immer im jeweiligen Kapitel angemerkt, ob die absolute Kanalnummer oder aber die relative Kanalnummer im Bezug zum

## 5. Messung und Auswertung

gemessenen Nullpunkt betrachtet wird. In den bisher gezeigten Formeln muss selbstverständlich der Abstand zum Bezugspunkt, also die relative Kanalnummer, verwendet werden.

Das Eigenrauschen wurde für zwei verschiedene Zeiten gemessen. Einmal für 300 und einmal für 1800 Sekunden.

Schaut man sich die Formel an, so sieht man, dass durch den Mittelwert  $\mu$  geteilt wird. Dies stellt ein Problem dar, wenn man das Eigenrauschen misst. Misst man dieses, so misst man genau den Bezugspunkt des ADCs (Also im Fall von relativen Kanalnummern den Kanal Null). Diesen darf man nicht in die Formel zum Abschätzen einsetzen, da man sonst durch Null teilen würde. Deshalb wurde hier zum abschätzen des Asymmetriewertes aufgrund des Eigenrauschens ein  $\mu$  gesetzt bei 5 V. Dies ist gerade in der Hälfte des dynamischen Bereichs von zehn Volt. Der Abstand von null Volt zu fünf Volt in Kanalnummern beträgt für die 40 Sample Einstellung: 2621576.

In Abbildung 25 sind die beiden Messergebnisse aufgetragen. Man sieht in der logarithmischen Darstellung bei beiden Messungen eine deutliche Schulter auf der linken Seite. Da es sich hierbei um das Eigenrauschen des ADCs handelt, kann diese nur durch die im ADC verbaute Elektronik hervorgerufen werden und scheint damit eine Eigenart des ADCs zu sein. An späterer Stelle sieht man, dass diese nur bei Spannungen  $> -5$  Volt zu sehen ist. Man sieht auch, dass die Schulter sehr viel kleiner ist als das Peak, welcher etwa 3 Größenordnungen über der Schulter liegt in der Häufigkeit der Einträge. Durch den großen Unterschied in der Höhe werden auch die Werte für Mittelwert und Standardabweichung nicht signifikant beeinflusst.

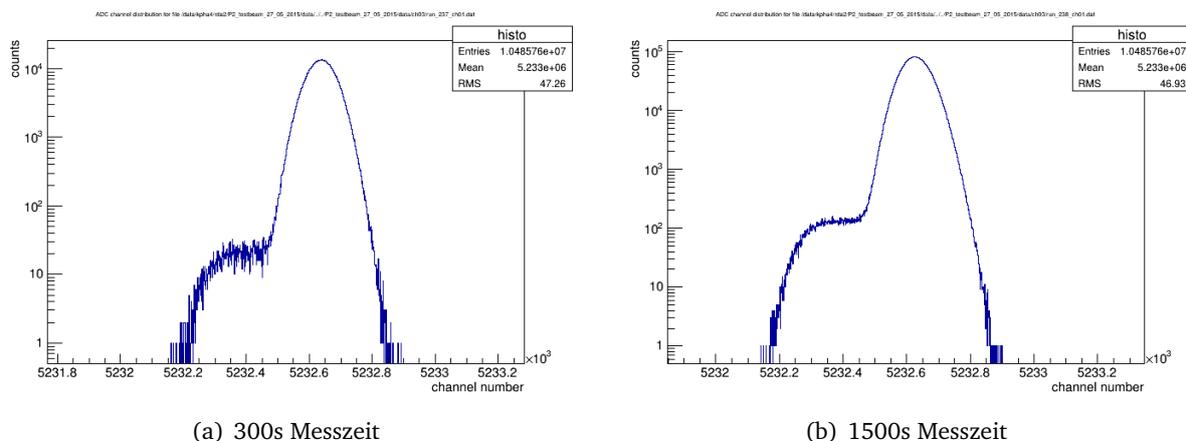


Abbildung 25: Aufnahme des Eigenrauschens mithilfe des Kurzschließens des ADC-Eingangs

Es wurde noch eine weitere Messung durchgeführt. Die ADCs wurden mit besonderen Vorverstärkern geliefert. Der Grund für diese Vorverstärker wird in einem späteren Kapitel erläutert. Da diese Verstärker auch später im Experiment zum Einsatz kommen sollen, wurde hier direkt noch das Rauschen der Messkette Vorverstärker-ADC gemessen. Dazu wurde auch hier der Vorverstärker mit seiner Betriebsspannung versorgt und dann der Eingang kurzgeschlossen, der Ausgang des Vorverstärkers auf den entsprechenden Kanal des ADCs gesteckt, und eine Messung gemacht. Die Ergebnisse dieser drei Messungen und die Abschätzung des Eigenrauschens finden sich in der nachfolgenden Tabelle.

## 5. Messung und Auswertung

---

Art	Messzeit /s	Sigma	$\Delta$ Asymmetrie	$\Delta$ Asymmetrie (10000 h)
ADC (Eigenrauschen)	1800	46.93	$1.46 \cdot 10^{-8}$	$4.22 \cdot 10^{-11}$
ADC (Eigenrauschen)	300	47.26	$6.01 \cdot 10^{-9}$	$4.25 \cdot 10^{-11}$
ADC+Vorverst.	60	39.27	$2.73 \cdot 10^{-8}$	$3.53 \cdot 10^{-11}$

Es wurde hier neben den direkt gemessenen Werten auch noch eine Extrapolation auf die Messgröße nach 10000 h in die Tabelle eingetragen. Man sieht, dass nach dieser Zeit der Beitrag durch das Eigenrauschen des ADCs hinreichend klein ist, sodass das Rauschen dieser Elektronik keine Behinderung für das Messen der Asymmetrie mit der gewünschten Genauigkeit darstellt. Auffällig ist, dass das Rauschen von ADC und Vorverstärker zusammen kleiner ist als nur das Rauschen des ADCs. Dies liegt aber wahrscheinlich an der doch sehr kurzen Messzeit von nur 60 Sekunden.

### 5.3. Linearität des TRIUMF ADCs

Um im späteren Experiment den gesamten dynamischen Bereich auszunutzen, wurde die Linearität des TRIUMF ADCs untersucht. Es wurde hier die integrale Nichtlinearität untersucht. Der Wert für die differentielle wurde dem Datenblatt entnommen.

Um die Linearität zu messen, wurde ein sogenannter Präzisions Spannungsgeber verwendet, welcher direkt mit einen ADC-Eingang verbunden war. Bei dem Gerät handelt es sich um einen ESTERLINE ANGUS V2000. Dieser wurde als Referenzspannungsquelle verwendet. Da das Gerät schon älter als 30 Jahre ist, gab es zu diesem Gerät kein Datenblatt. Auch bei dem Hersteller ist dieses Gerät nicht mehr erhältlich. Es wurde jedoch mithilfe eines PREMA 5017 Voltmeter festgestellt, dass das Gerät nicht driftet und bei einer Integrationszeit im Voltmeter über fünf Sekunden die Schwankungen des Spannungsnormals unter  $10 \mu\text{V}$  liegen. Da die Kanalbreite schon  $76.29 \mu\text{V}$  beträgt, stellen also zumindest langsame Schwankungen oder Drifts kein Problem dar. Schnelle Schwankungen des Spannungsgebers müssten durch die Verbreiterung der Gaußglocken oder durch nicht mehr gaußförmige Messergebnisse auffallen (deutliche Verbreiterung der Gaußglocke im Gegensatz zur Breite des Eigenrauschens).

Es wurde von -8 bis 8 Volt in 1 Volt Schritten gemessen. Wieder wurde mit einer Triggerfrequenz von 10 kHz gearbeitet. Die Messzeit für jeden Schritt beträgt 300 Sekunden.

Zwei Beispiele solch einer Messung sind in Abbildung 26 angegeben.

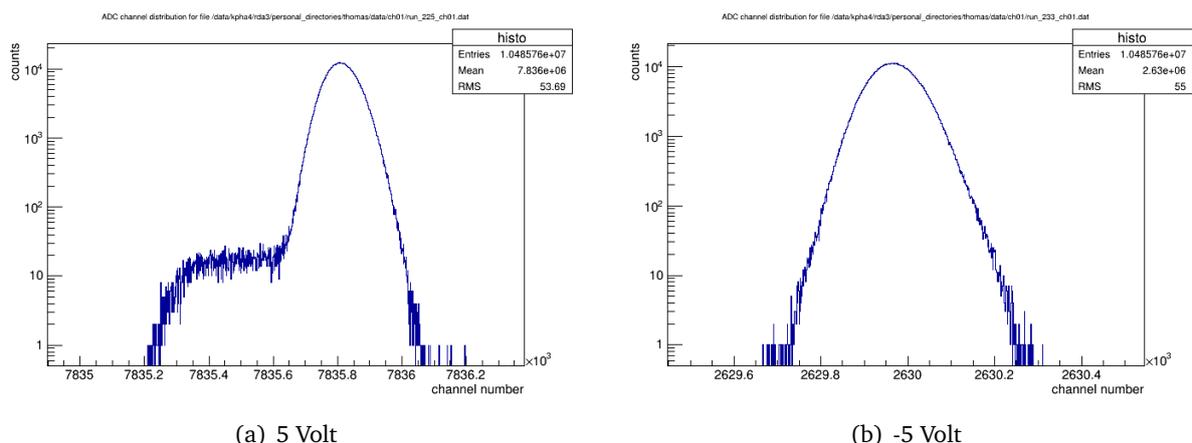


Abbildung 26: Logarithmisch aufgetragene Messergebnisse mit ESTERLINE ANGUS V2000 als Spannungsquelle bei einer Messzeit von 300 Sekunden. Bei Abb. 26(a) ist eine klare Schulter zu sehen, während bei Abb. 26(b) ein fast symmetrisches Ergebnis zu sehen ist.

Man sieht bei der positiven Spannung eine Schulter an der Gaußglocke auf der linken Seite. Diese ist bei allen Messungen im positiven Spannungsbereich ausgeprägt. Sie nimmt im negativen Spannungsbereich ab. Wie man sieht ist die Schulter dann bei etwa -5 Volt komplett verschwunden, und es ergibt sich eine symmetrische Verteilung. Die Schulter hat keinen nennenswerten Einfluss auf die Bestimmung des Mittelwertes und der Breite der Verteilung, da sie zwei bis drei Größenordnungen unter dem Peak liegt. Die Schulter ist nur aufgrund der logarithmischen Darstellung sichtbar. Man sieht jedoch an der Breite der Verteilung, da diese nur etwas breiter

## 5. Messung und Auswertung

---

(Vergleiche Breite ca. 55 Kanäle, mit Eigenrauschen von 47) ist als das Eigenrauschen, dass die Spannungsquelle eine gute konstante Spannung liefert.

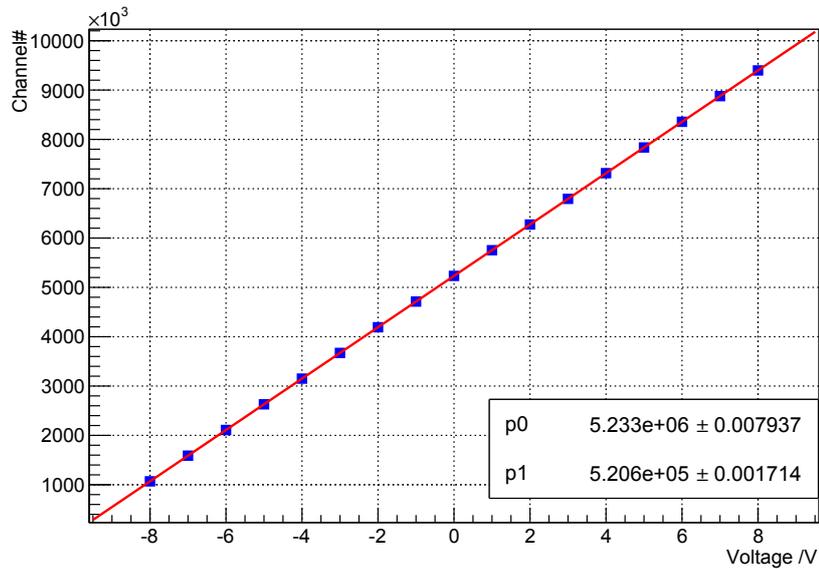
Es wurde nun aus jedem dieser Messergebnisse der Mittelwert und die Breite bestimmt. Die Ergebnisse sind in der nachfolgenden Tabelle aufgetragen.

Spannung /V	$\mu_{\text{absolut}}$	$\sigma$	$\Delta\mu$
8	9397673.68	70.95	0.0410
7	8877113.64	58.34	0.0337
6	8356420.44	65.03	0.0375
5	7835811.77	53.69	0.0310
4	7315195.31	59.97	0.0346
3	6794563.32	62.64	0.0362
2	6273983.13	64.01	0.0370
1	5753339.24	56.33	0.0325
0	5232653.46	42.34	0.0244
-1	4712274.06	56.34	0.0325
-2	4191625.48	53.16	0.0307
-3	3671152.68	56.55	0.0326
-4	3150571.42	52.46	0.0303
-5	2629969.33	55.00	0.0318
-6	2109449.07	50.58	0.0292
-7	1588956.90	56.94	0.0329
-8	1068433.84	67.5	0.0390

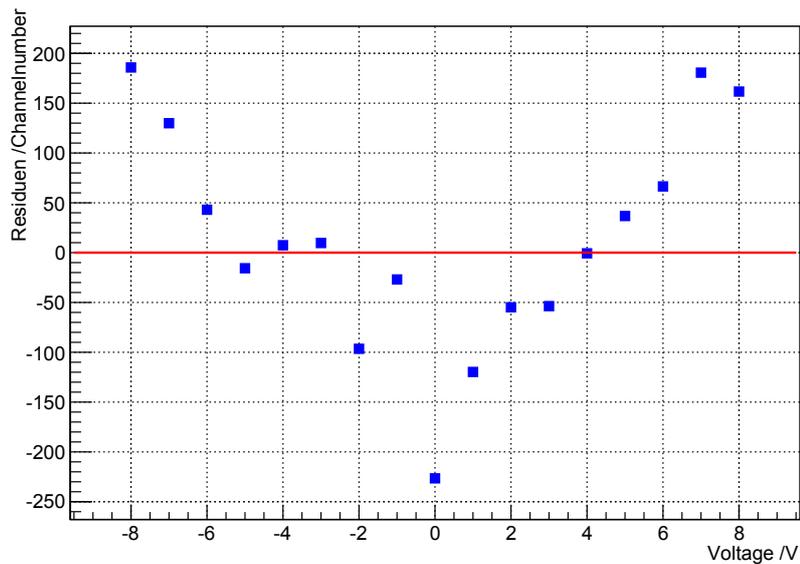
An dieser Stelle ist zu beachten, dass  $\mu$  die absoluten Werte der Kanalnummer sind und nicht der Abstand zum Punkt 0 Volt. Das ist an dieser Stelle auch nicht nötig, da man hier nicht mit Gleichungen arbeitet, die dies benötigen. Desweiteren wird bei den nachfolgenden Residuen-Plots der Fit von den Daten abgezogen. So ergeben sich auch hier wieder die relativen Kanalnummern.

Man sieht, dass  $\sigma$  größer wird für größere Spannungen, sowohl im negativen als auch positiven Spannungsbereich. Der Grund dafür ist sicher die größere Last unter der das Spannungsnormale bei einer höheren anliegenden Spannung steht. Der Fehler des Mittelwerts berechnet sich auch hier über die Formel  $\sigma/\sqrt{M}$ , wobei  $M$  die Anzahl der Messungspunkte ist.

Die Ergebnisse wurden in Abbildung 27(a) geplottet und gefittet. Die Fehler sind mit eingetragen, aber nicht zu erkennen aufgrund ihrer geringen Größe.



(a) Linearitäts Plot



(b) Residuen Plot aus linearem Fit

Abbildung 27: Überprüfen der Linearität des ADCs

Hier sieht man schon einmal in groben Zügen, dass der Fit und die Daten gut übereinstimmen. Die Fitgleichung war in diesem Fall:

$$f(u) = p0 + p1 \cdot u \quad (26)$$

## 5. Messung und Auswertung

Die Parameter haben trotz ihrer Größe ( $10^5$  und  $10^6$ ) eine hohe Genauigkeit. Ihr relativer Fehler liegt bei nur  $\Delta p_0/p_0 \approx 1.5 \cdot 10^{-9}$  und  $\Delta p_1/p_1 \approx 3.3 \cdot 10^{-9}$ . Lediglich das hohe  $\chi^2 = 6.8 \cdot 10^6$  deutet darauf hin, dass eine falsche Fitfunktion gewählt wurde oder aber die Fehler zu klein gewählt wurden.

Dies ist aber nicht weiter verwunderlich, da, wie im Abschnitt 4 schon beschrieben, die Umsetzungsfunktion eines ADC nie wirklich linear ist. Sie sollte nur möglichst gut einer Gerade entsprechen. Um dies genauer zu beurteilen, zieht man den Fit von den Daten ab. Mithilfe des so erstellten Residuen-Plots lässt sich die Linearität deutlich besser beurteilen als dies mit dem normalen Fit möglich wäre. Dies ist in Abbildung 27(b) aufgetragen.

Man sieht, dass die Messwerte nicht ganz mit der linearen Vorhersage übereinstimmen. Auch hier sind die Fehler mit eingezeichnet, sie sind jedoch auch hier zu klein, um sie sehen zu können. Man muss jedoch sagen, dass die Kanalnummern hier im Bereich  $10^2$  neben dem linearen Verlauf liegen, während man in Abbildung 27(a) Kanalnummern im Bereich  $10^6$  zu finden sind. Die Abweichung von der Linearität fällt also sehr klein aus. Dennoch scheint es eine Struktur in der Abweichung zu geben. Die Messdaten sind nicht einfach statistisch um die Messwerte verteilt. Es sieht ähnlich zu einer Parabel aus. Das heißt bei einem höheren Spannungsbetrag fallen die getroffenen Kanäle zu hoch aus, bei kleinen eher zu niedrig.

Um dies zu überprüfen, wurde wieder ausgehend von den Daten ein weiterer Fit erstellt, diesmal jedoch mit einem Polynom zweiten Grades.

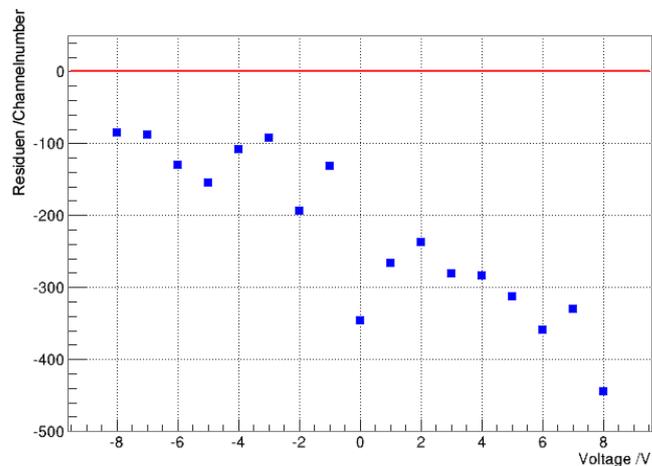


Abbildung 28: Residuen Plot aus Fit mit Polynom zweiten Grads

Das Ergebnis hat ein besseres  $\chi^2$  von  $3.1 \cdot 10^5$ , welches immer noch sehr groß ist, aber Faktor 20 kleiner als das des linearen Fits. Das Ergebnis, welches in Abbildung 28 zu sehen ist, überzeugt hingegen nicht. Alle Werte liegen unter dem Fit.

Die größten Abstände sind nun auch fast doppelt so hoch wie man leicht an der Y-Achse sieht. Man kann sehen, dass die Werte nun anhand einer fallenden Gerade verlaufen. Um dies auszugleichen hätte nur die Steigung beim Polynom Fit kleiner ausfallen müssen. Dennoch scheint mit diesem Ergebnis das  $\chi^2$  am kleinsten zu sein. Da dieses Ergebnis augenscheinlich aber schlechter passt, als das des linearen Fits, wurde noch eine letzte Methode verwendet. Bei dieser wurde

## 5. Messung und Auswertung

das Ergebnis aus Abbildung 27(b) noch einmal mit nur einer Parabel Gleichung angefitet.

$$f(u) = p_2 \cdot u^2 + p_3 \quad (27)$$

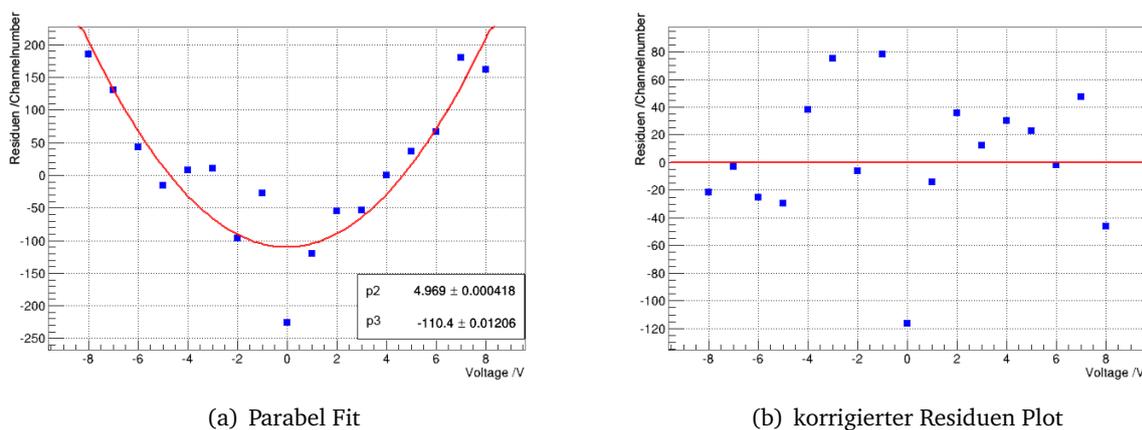


Abbildung 29: Sequenzielles Fitten, im linken Bild sieht man den Parabel Fit des Residuen-Plots aus Abb. 27(b). Im rechten den daraus korrigierten Residuenplot.

Das Ergebnis des Fits sehen wir in Abbildung 29(a) den daraus berechneten zweiten korrigierten Residuen Plot rechts daneben in Abbildung 29(b).

Wie man sieht, passt dieser Residuen Plot nun deutlich besser zu den Daten, als der der aufgrund des Polynom zweiten Grades direkt erstellt wurde. Man sieht also, dass die Linearität des des ADCs gut gegeben ist, auch wenn es eine leichte quadratische Struktur darauf zu geben scheint. Man sieht es am Parameter der  $p_2$  in Abbildung 29(a), welcher die Amplitude des quadratischen Terms darstellt. Dieser ist sehr klein im Gegensatz zur Amplitude der Geradensteigung aus 27(a) (Amplitude 4.969 gegenüber  $5.206 \cdot 10^5$ ). Auffällig ist, dass egal bei welcher Art des Fits, der Punkt bei genau null Volt Eingangsspannung der ist, der am meisten neben den Fits liegt. Einen Offset des Spannungsnormals kann ausgeschlossen werden, da man genau diesen aufgrund der Messung bei null Volt korrigiert hätte.

Da die quadratische Struktur allerdings nur sehr klein ist und die Residuen für einen einfachen linearen Fit nur maximal etwa 200 Kanäle groß sind, was bei Kanalnummern in der Größenordnung  $10^5$  relativ und  $10^6$  absolut einen verschwindend kleinen Anteil darstellt, ist das ADC in guter Näherung ein lineares Messgerät. Dadurch kann später im Experiment ein großer dynamischer Bereich benutzt werden, ohne Korrekturen berechnen zu müssen.

### 6. Strahlzeit

Während dieser Arbeit gab es zwei Teststrahlzeiten am Elektronenbeschleuniger MAMI. Es war also möglich die Elektronik an einem Elektronenstrahl zu testen. Der Strahl war nicht polarisiert. Deshalb gibt es keine verschiedenen Helizitäten, aber wie später am richtigen P2-Experiment wird der Strahl mithilfe von Quarzstäben in ein Lichtsignal umgewandelt, was dann mit der PMT detektiert werden kann.

#### 6.1. Strahlzeit 13.3.2015

Während dieser Strahlzeit wurde das Verhalten des Signals bei verschiedenen Raten gemessen. Die Raten selbst wurden dabei mit Plastiksintillatoren bestimmt, die in Strahlrichtung nach dem Experiment kamen.

Man erwartet einen linearen Zusammenhang zwischen der Erhöhung der Strahlrate und dem Ansteigen der Kanalnummer im ADC. Der Messaufbau hat das folgende Schema:

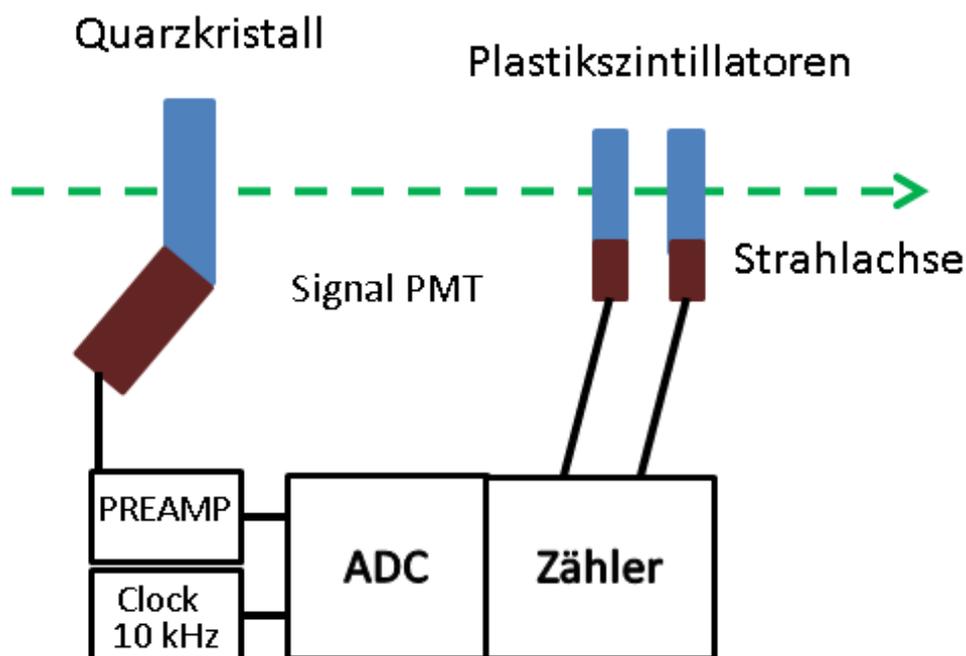


Abbildung 30: Schematischer Messaufbau der Strahlzeit

Der Strahl des Beschleunigers trifft auf einen Quarzstab. Er erzeugt in dem Quader Cherenkov-Licht, welches durch den PMT in ein elektrisches Signal umgewandelt wird. Das Signal des PMTs kommt in einen Vorverstärker. Der Grund für den Vorverstärker ist folgender:

Der erste Grund ist, dass ein PMT eine typische Stromquelle ist. Auf der Anode kommen eine gewisse Anzahl Elektronen pro Zeiteinheit an, welche dann zum Verstärker fließen. Bei dem Verstärker kann man nun über einen Schalter einstellen, durch welchen Widerstand der Strom fließen soll. Somit bestimmt man, welche Spannung abfällt. Dies stellt eine Verstärkung und eine Impedanzwandlung dar. Ein Beispiel:

## 6. Strahlzeit

---

Von der Anode des PMTs fließen  $1 \mu\text{A}$  ab. Fließen diese durch einen Widerstand von  $10 \text{ k}\Omega$  so ergibt sich ein Spannungsabfall von  $0.01 \text{ Volt}$ . Fließt der gleiche Strom durch einen Widerstand von  $1 \text{ M}\Omega$ , so fällt eine Spannung von  $1 \text{ Volt}$  ab.

Das ist von Vorteil, da das ADC nur einen festen Eingangswiderstand hat und man somit keine Möglichkeit hätte, bei unterschiedlichen Raten die Spannung so einzustellen, dass sie im dynamischen Bereich des ADCs liegt.

Der zweite Grund für die Notwendigkeit des Vorverstärkers ist, dass der Signalpuls eines PMTs eine Breite von etwa  $20 \text{ ns}$  besitzt. Das ADC kann jedoch nach dem Abtasttheorem nur Signale verlustfrei digitalisieren, wenn die Bandbreite des Signals halb so hoch ist wie die Abtastrate des ADCs. Das ADC hat eine maximale Abtastrate von  $500 \text{ kSp/s}$ . Dies bedeutet, dass das Signal keine Frequenz höher als  $250 \text{ kHz}$  beinhalten darf. Damit ergibt sich eine Periodendauer des Signals von mindestens  $4 \mu\text{s}$ . Mit einer Signalbreite von  $20 \text{ ns}$  ist dies deutlich unterschritten. Der Vorverstärker dient deshalb dazu, diese schnellen Signale aufzuintegrieren. Die Vorverstärker selbst besitzen eine Bandbreite von  $16 \text{ kHz}$ . Dies bedeutet, dass alle Signale, die schneller sind als  $16 \text{ kHz}$  von dem Verstärker zu einer Gleichspannung integriert werden. Aus den vielen schnellen Pulsen die sich aus dem PMT überlagern, wird also nun eine Gleichspannung. Diese kann nun mit dem ADC digitalisiert werden.

Wenn die Rate des Strahls konstant ist, erwartet man auf den Spektren Gaußverteilungen um den wahren Spannungswert.

Aufgrund der begrenzten Zeit konnten nur jeweils drei Messungen (Rate auf den Plastikszintillatoren:  $1.2 \text{ MHz}$ ,  $3.5 \text{ MHz}$  und  $5.6 \text{ MHz}$ ) mit drei verschiedenen Messzeiten ( $300$ ,  $600$ ,  $900 \text{ Sekunden}$ ) bei zwei verschiedenen Versorgungsspannungen gemacht werden. Dazu kommt noch jeweils eine Untergrundmessung. Es ist an dieser Stelle noch anzumerken, dass der Strahl bei dieser Strahlzeit nicht gut stabil war, sodass Schwankungen in der Rate von ca.  $10\%$  der Gesamtrate abgeschätzt wurden.

Um sich das Ganze noch etwas besser klar zu machen, sieht man in Abbildung ?? auch noch einmal ein Foto des Aufbaus. Dabei ist anzumerken, dass man hier nur den Quarzstab mit dem PMT sehen kann, die Plastikszintillatoren und das Rack mit ADC und Zähler ist auf den Foto nicht enthalten.

## 6. Strahlzeit

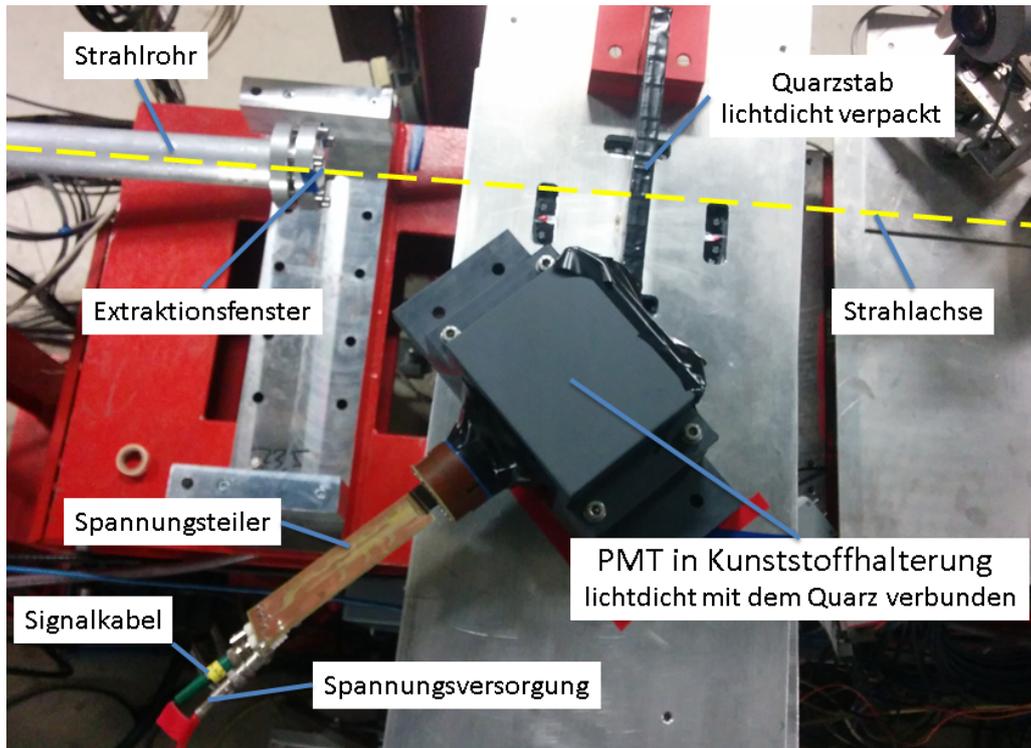


Abbildung 31: Foto des Messaufbaus der Strahlzeit

Beispielhaft ist in Abbildung 32 einmal das Signal des Untergrunds und bei einer Rate von 3.5 MHz bei 1000 Volt Versorgungsspannung zu sehen (300s Messzeit). Man sieht bei der Aufnahme des Untergrunds in 32(a) wie auch schon bei der Messung des Eigenrauschens die Schulter auf der linken Seite des Peaks. Sie erscheint hier nur schmaler, weil die Skala größer ist, als in den Plots aus Kapitel 5.2.

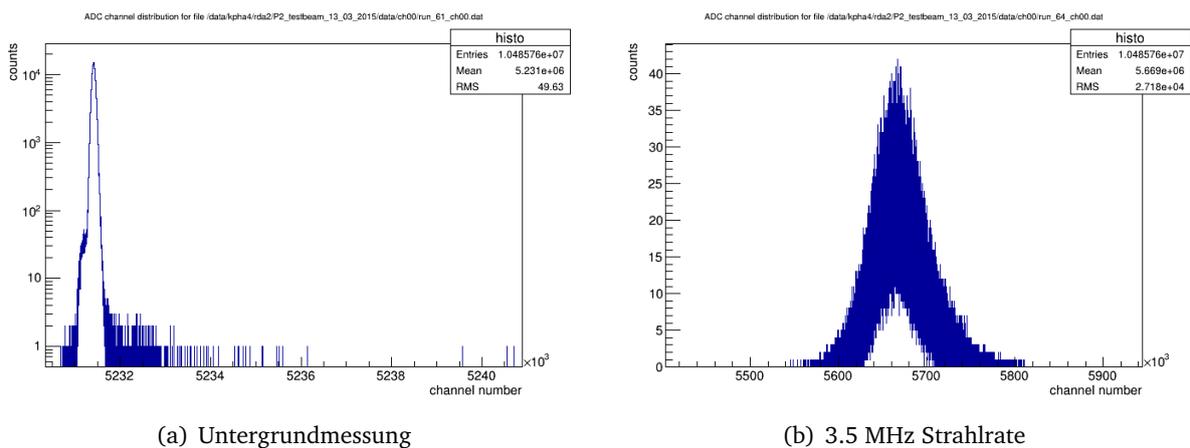


Abbildung 32: Ergebnisse der Strahlzeit bei 1000 Volt und 300 Sekunden Messzeit

Der Grund, weswegen die Skala hier größer ist, sind die Ereignisse neben den Peaks, welche nur vereinzelt vorkommen. Diese gehören nicht zum Rauschen des ADCs, sondern sind der Untergrund zusammengesetzt aus Dunkelpulsen des PMTs und nicht 100% lichtdicht abgeschirmten PMT. Trotzdem ist die Breite des Peaks kaum beeinflusst, sodass bei der Untergrund Messung immer noch Peakbreiten von nur knapp 50 Kanälen gemessen werden. Neben dem Untergrund sieht man in Abbildung 32(b) die Messung für 3.5 MHz. Dieser Peak ist sehr viel breiter als der des Untergrunds. Durch diese Verbreiterung des Peaks ist nun auch die Schulter auf der linken Seite überlagert und nicht mehr sichtbar.

Der Grund dafür, dass der Peak diese große Breite aufweist ist folgender: Einmal ändert sich  $\mu$  aufgrund der Ratenschwankung welche hier sehr groß war. Der zweite und wichtigste Grund für die Breite der Peaks ist das Schrotrauschen. Dieses tritt an jeder Stelle der Elektronik auf, an der die Elektronen eine Potentialbarriere überwinden müssen. Die für uns interessante Stelle ist am Punkt der Anode, da hier der höchste Strom fließt. Das Rauschen folgt dabei der Formel:

$$\sigma_{shot} = \sqrt{2 \cdot q \cdot I \cdot B} \quad (28)$$

Hierbei ist  $q$  die Ladung, die an der Anode aus einem Elektron ankommen würde,  $I$  der Strom, der bei der Messung an der Anode anlag, und  $B$  die Bandbreite des Signals.

Wir berechnen die doppelte Bandbreite aus der Messzeit jedes Helizitätsfensters. Da wir die gleichen Einstellungen getroffen haben, wie bei der Messung des Eigenrauschens nehmen wir 40 Samples auf, was bei 500 kSp/s, einer Messzeit von 80  $\mu$ s entspricht. Also ist  $B' = 2B = (80 \mu\text{s})^{-1} = 12.5 \text{ kHz}$ .

Um  $q$  zu bestimmen, muss man aus den Daten die Verstärkung des PMTs bei der jeweiligen Versorgungsspannung berechnen. Dies geschieht, indem man die Rate an Teilchen bestimmt, die aus der Anode fließen und diese mit der Rate vergleicht, die aus dem Plastikszintillator abzulesen ist. Kennt man die Verstärkung, ergibt sich  $q$  aus der Multiplikation der Anzahl an Photonen, die der Quarzstab erzeugt, dem Anteil der nach der Erzeugung an der PMT ankommt, der Verstärkung des PMTs und der Elementarladung.

$I$  wird bestimmt, indem man von der gemessenen Kanalnummer zurückrechnet auf den Strom der aus der Anode geflossen sein muss, um die entsprechende Kanalnummer zu treffen.

Das so bestimmte  $\sigma_{shot}$  ist in der Einheit Ampere und muss deshalb noch in die Größe Kanalnummer umgerechnet werden, um es mit der gemessenen Gesamtbreite zu vergleichen.

Nun kann man die Gesamtbreite mit dem berechneten Schrotrauschen entfalten. Aus der so verbliebenen Breite kann man sehen, ob diese Ursache tatsächlich den Hauptteil der Gesamtbreite ausmachen. Die verbleibende Breite des Peaks ist dann hauptsächlich von der Ratenschwankung abhängig. Dies wird im nächsten Kapitel bei einer zweiten Strahlzeit noch genauer untersucht, da dort die Rate stabil war.

In Abbildung 33 sieht man nun die Messungen für alle drei Messzeiten bei einer Betriebsspannung von 1000 Volt eingetragen und gefittet.

## 6. Strahlzeit

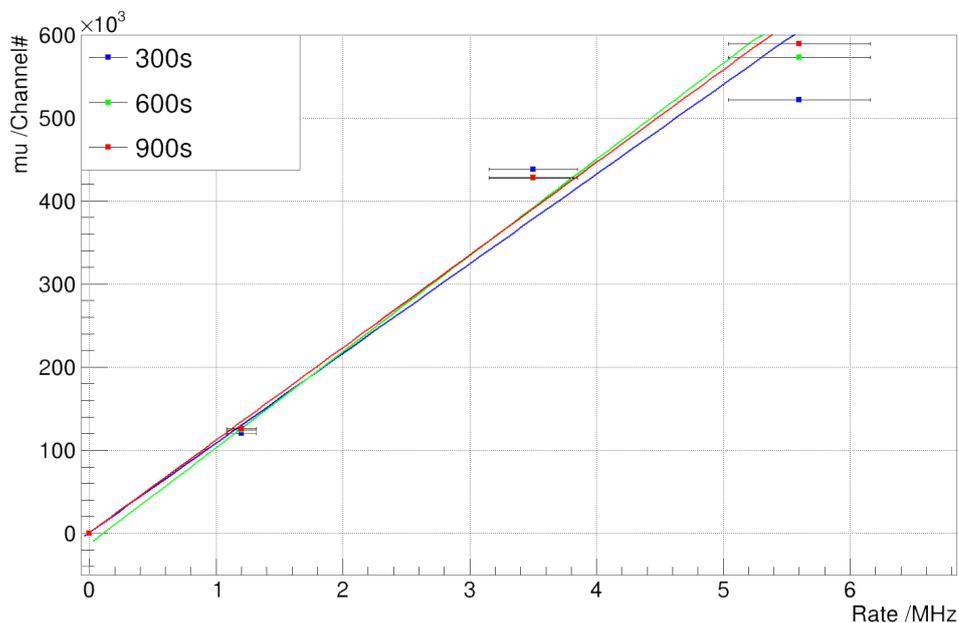


Abbildung 33: Messergebnisse der Strahlzeit bei 1000 Volt Versorgungsspannung

Bei dem Graph sind die relativen Kanalnummern zu sehen. Es ist auffällig, dass von der Fitgeraden aus gesehen, jeweils der Wert für 1.2 MHz zu niedrig, der Wert bei 3.5 MHz zu hoch und der Wert bei 5.6 MHz wieder zu niedrig gemessen wird. Es wurde anhand folgender Gleichung gefittet:

$$f(\nu) = m \cdot \nu + b \quad (29)$$

Es ergeben sich folgende Parameter für den Fit:

Messzeit /s	Parameter		
	$m$	$b$	$\chi^2/ndf$
300	$1.079 \cdot 10^5 \pm 6281$	$-2.928 \cdot 10^{-8} \pm 0.02084$	4.942/2
600	$1.157 \cdot 10^5 \pm 1.220 \cdot 10^4$	$-1.347 \cdot 10^4 \pm 2.344 \cdot 10^4$	1.679/1
900	$1.114 \cdot 10^5 \pm 6450$	$-9.100 \cdot 10^{-9} \pm 0.0175$	1.597/2

Bei der Messung mit 600 Sekunden Messzeit gibt es einen Freiheitsgrad weniger, da dort die Messung für den Untergrund fehlerhaft gespeichert wurde (Datei unlesbar) und deshalb im Fit der Nullpunkt nicht berücksichtigt wurde. Dies sieht man auch am Parameter  $b$ , welcher um viele Größenordnungen größer ist als bei den anderen. Man berechnet nun das Schrotrauschen und entfaltet es mit der Gesamtbreite. Die Ergebnisse sind in der nachfolgenden Tabelle eingetragen.

## 6. Strahlzeit

1000 Volt Versorgungsspannung, high Gain					
Messzeit /s	Rate /MHz	$\mu_{rel}$	$\sigma_{gem}$	$\sigma_{shot}$	$\sigma_{rest}$
300	Background	$0 \pm 0.029$	49.63	0	49.63
300	$1.2 \pm 0.12$	$1.20 \cdot 10^5 \pm 7.14$	$1.24 \cdot 10^4$	$1.22 \cdot 10^4$	$1.74 \cdot 10^3$
300	$3.5 \pm 0.35$	$4.38 \cdot 10^5 \pm 15.69$	$2.72 \cdot 10^4$	$2.62 \cdot 10^4$	$7.32 \cdot 10^3$
300	$5.6 \pm 0.56$	$5.22 \cdot 10^5 \pm 17.42$	$3.02 \cdot 10^4$	$2.47 \cdot 10^4$	$1.74 \cdot 10^4$
600	Background	0	-	0	-
600	$1.2 \pm 0.12$	$1.24 \cdot 10^5 \pm 5.18$	$1.27 \cdot 10^4$	$1.27 \cdot 10^4$	$1.06 \cdot 10^3$
600	$3.5 \pm 0.35$	$4.27 \cdot 10^5 \pm 10.89$	$2.67 \cdot 10^4$	$2.55 \cdot 10^4$	$7.79 \cdot 10^3$
600	$5.6 \pm 0.56$	$5.73 \cdot 10^5 \pm 12.04$	$2.95 \cdot 10^4$	$2.71 \cdot 10^4$	$1.17 \cdot 10^4$
900	Background	$0 \pm 0.018$	52.62	0	52.62
900	$1.2 \pm 0.12$	$1.26 \cdot 10^5 \pm 4.28$	$1.29 \cdot 10^4$	$1.29 \cdot 10^4$	$5.02 \cdot 10^2$
900	$3.5 \pm 0.35$	$4.28 \cdot 10^5 \pm 8.87$	$2.66 \cdot 10^4$	$2.37 \cdot 10^4$	$7.34 \cdot 10^3$
900	$5.6 \pm 0.56$	$5.89 \cdot 10^5 \pm 10.14$	$3.04 \cdot 10^4$	$2.78 \cdot 10^4$	$1.23 \cdot 10^4$

Man sieht hier eine deutliche Verkleinerung des Sigmas nach dem Entfalten. Auffällig ist, dass die Werte von  $\sigma_{shot}$  bei kleinen Raten sehr nahe an  $\sigma_{gem}$  sind. Für die höheren Raten driftet es allerdings weiter auseinander, was man auch an den stark steigenden Werten von  $\sigma_{rest}$  ersichtlich wird. Dies ist bei allen Messzeiten der Fall. Wenn man überlegt, dass die Ratenschwankung aber auch mit der Rate steigt, so müsste diese einen immer größeren Anteil an der Gesamtbreite ausmachen. Man sieht auch hier, dass bei der Messung für 600 Sekunden kein gemessenes Sigma angegeben wird, da dort die Daten unbrauchbar waren.

Analog wird nun die gleiche Betrachtung für eine Versorgungsspannung von 1200 Volt durchgeführt. In Abbildung 34 sieht man die Ergebnisse aufgetragen.

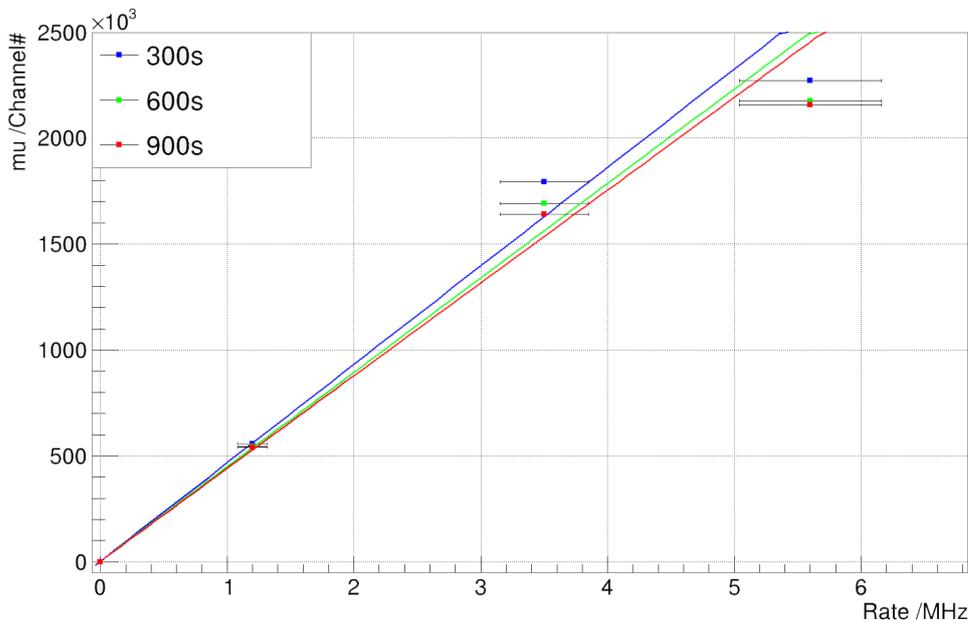


Abbildung 34: Messergebnisse der Strahlzeit bei 1200 Volt Versorgungsspannung

## 6. Strahlzeit

Die Fitgleichung ist wie vorher auch eine Gerade:

$$f(\nu) = m \cdot \nu + b \quad (30)$$

Messzeit /s	Parameter		
	$m$	$b$	$\chi^2/ndf$
300	$4.648 \cdot 10^5 \pm 2.696 \cdot 10^4$	$2.524 \cdot 10^{-9} \pm 0.0652$	2.676/2
600	$4.469 \cdot 10^5 \pm 2.586 \cdot 10^4$	$1.167 \cdot 10^{-8} \pm 0.003$	2.403/2
900	$4.380 \cdot 10^5 \pm 2.537 \cdot 10^4$	$8.580 \cdot 10^{-9} \pm 0.0315$	2.056/2

In der Grafik sind die gleichen Charakteristika wie bei einer Versorgungsspannung von 1000 Volt zu sehen. Lediglich die Steigung der Geraden ist deutlich höher. Dies ist sofort durch die höheren Versorgungsspannung und der deshalb höheren Verstärkung des PMTs erklärbar. Auch hier wurde anhand der Daten das Schrotrauschen mit den Messungen entfaltet, um so zu sehen, wieviel des Gesamtrauschens verbleibt.

1000 Volt Versorgungsspannung, high Gain					
Messzeit /s	Rate /MHz	$\mu_{rel}$	$\sigma_{gem}$	$\sigma_{shot}$	$\sigma_{rest}$
300	Background	$0 \pm 0.07$	125.7	0	125.7
300	$1.2 \pm 0.12$	$5.56 \cdot 10^5 \pm 32.74$	$5.67 \cdot 10^4$	$5.62 \cdot 10^4$	$7.26 \cdot 10^3$
300	$3.5 \pm 0.35$	$1.80 \cdot 10^6 \pm 66.11$	$1.15 \cdot 10^5$	$1.04 \cdot 10^5$	$4.76 \cdot 10^4$
300	$5.6 \pm 0.56$	$2.27 \cdot 10^6 \pm 74.02$	$1.28 \cdot 10^5$	$1.05 \cdot 10^5$	$7.35 \cdot 10^4$
600	Background	$0 \pm 0.05$	120.6	0	120.6
600	$1.2 \pm 0.12$	$5.46 \cdot 10^5 \pm 22.61$	$5.54 \cdot 10^4$	$5.57 \cdot 10^4$	$6.20 \cdot 10^3$
600	$3.5 \pm 0.35$	$1.69 \cdot 10^6 \pm 42.25$	$1.04 \cdot 10^5$	$1.01 \cdot 10^5$	$2.21 \cdot 10^4$
600	$5.6 \pm 0.56$	$2.18 \cdot 10^6 \pm 43.07$	$1.06 \cdot 10^5$	$1.03 \cdot 10^5$	$2.37 \cdot 10^4$
900	Background	$0 \pm 0.04$	115.7	0	115.7
900	$1.2 \pm 0.12$	$5.41 \cdot 10^5 \pm 18.30$	$5.49 \cdot 10^4$	$5.52 \cdot 10^4$	$5.80 \cdot 10^3$
900	$3.5 \pm 0.35$	$1.64 \cdot 10^6 \pm 33.13$	$9.94 \cdot 10^4$	$9.81 \cdot 10^4$	$1.58 \cdot 10^4$
900	$5.6 \pm 0.56$	$2.16 \cdot 10^6 \pm 34.63$	$1.04 \cdot 10^5$	$1.02 \cdot 10^5$	$2.05 \cdot 10^4$

Man sieht, dass bei dieser Versorgungsspannung ein deutlich höherer Untergrund messbar ist. Dies ist auf die erhöhte Lichtempfindlichkeit des PMTs zurückzuführen. Somit wirken sich Lichtlecks deutlich stärker aus und Dunkelpulse werden häufiger. Generell sind die Fehler höher als bei der Messung mit 1000 Volt Betriebsspannung. Auch hier ist  $\sigma_{shot}$  nahezu gleich dem  $\sigma_{gem}$  für die kleinen Strahlraten. Bei 600 und 900 Sekunden Messzeit bei einer Rate von 1.2 MHz, ist  $\sigma_{shot}$  sogar größer als das gemessene Sigma. Der Grund hierfür könnte sein, dass die Rate von den Plastikszintillatoren zu niedrig angezeigt wurde, somit bestimmt man die Verstärkung des PMTs zu hoch und erhält einen zu großen Wert für  $\sigma_{shot}$ .

Es verbleibt bei beiden Versorgungsspannungen vor allem bei höheren Raten ein hohes  $\sigma_{rest}$ . Ein Grund dafür ist sicher die Breite die durch die Ratenschwankung entstanden ist. Diese herauszurechnen gestaltet sich als eher schwierig, da die Schwankung in der Rate bei dieser Strahlzeit nur abgeschätzt wurde. Ein weiterer Grund könnten die falsch bestimmten Raten sein, vor allem bei den höheren Raten. Der Puls in den Plastikszintillatoren hat etwa die Länge von 20 ns. Dies

wird in einen NIM-Puls umgewandelt und dann in einen Counter geführt. Der NIM-Puls hat eine Breite von 55 ns für den ersten Plastiksintillator und eine Breite von 25 ns für den zweiten Plastiksintillator (Der Grund dafür liegt in einer Koinzidenzschaltung für ein anderes Experiment während der Strahlzeit). Geht man nun beispielhaft von einer Rate von 10 MHz aus, so ist der mittlere Abstand zwischen zwei Pulsen 100 ns. Da die Pulse aber nicht in ihrem mittleren Abstand aufeinander folgen, spielt die Totzeit eine nicht unerhebliche Rolle. Diese Besonderheit wird bei den Ergebnissen aus der zweiten Strahlzeit sehr deutlich. Da hier die Raten aber unzureichend genau bestimmt wurden und auch die Schwankungen nur abgeschätzt wurden, wurde bei diesen Tests darauf verzichtet diese Totzeit zu korrigieren. Bei der genaueren Analyse des Verhaltens der Rate in der zweiten Strahlzeit, ist diese Korrektur durchgeführt worden.

Abschließend ist zu dieser Strahlzeit zu sagen, dass es der erste Test der teils selbstentwickelten Elektronik an einem Elektronenstrahl war. Alle Geräte haben gut funktioniert, lediglich das Umschalten der Verstärkung konnte bei dieser Strahlzeit noch nicht getestet werden, da der Zeitplan dies nicht zuließ.

## 6.2. Strahlzeit 27.5.2015

Der Messaufbau ist der gleiche wie bei der Strahlzeit aus Kapitel 6.1. Diesmal waren allerdings die Raten sehr stabil, die vom Beschleuniger auf unseren Quarzstab geschickt wurden. Des weiteren wurde diesmal die Rate der beiden Plastikszintillatoren aufgezeichnet, sodass keine Abschätzung zu den Fehlern in der Rate gemacht werden muss, sondern sich dieser aus der Statistik ergibt. Es wurde hier einmal die gleiche Messung gemacht wie bei der ersten Strahlzeit, also aufgezeichnet, wie das Signal sich bei verschiedenen Strahlraten verhält. Dann wurde auch wieder die Gesamtbreite mit dem Schrotrauschen entfaltet, um zu sehen wie sich die verbleibende Breite verhält. Bei dieser Strahlzeit wurde nur für eine Messzeit und eine Versorgungsspannung gemessen. Dafür gibt es mit der Untergrund-Messung nun 20 Messpunkte auf der Kurve, sodass sich ein Verhalten besser als mit nur vier Messpunkten beurteilen lässt. Die jeweilige Rate wurde aus dem Zähler, mit dem die beiden Plastikszintillatoren verbunden waren, ausgelesen. Es ergeben sich deshalb für jeden Datenpunkt zwei verschiedene Raten. Der Einfachheit halber wird ab jetzt der erste Plastikszintillator Trigger1 und der zweite Trigger2 genannt (in Tabellen zum Teil auch T1 und T2 aus Platzgründen). In der Abbildung 35 sehen wir nun die Ergebnisse dieser Strahlzeit. Aufgetragen sind die Mittelwerte der bei jeder Rate gemessenen Häufigkeitsverteilungen mit ihren Messfehlern.

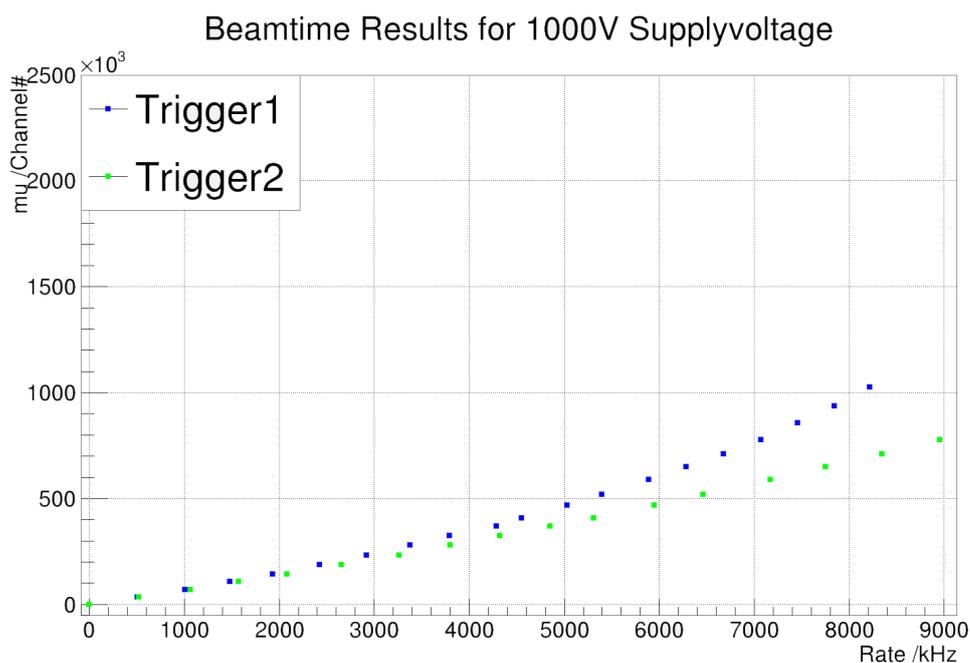


Abbildung 35: Mittlere getroffene Kanalnummer in Abhängigkeit der Triggerrate. Trigger1 ist in blau, Trigger2 in grün. Aufgrund der unterschiedlichen Totzeiten sind beide Kurven unterschiedlich stark gekrümmt und zueinander verschoben.

Auf der Abbildung kann man erstens sehen, dass beide Trigger für die gleiche Strahlstromeinstellung verschiedene Raten messen. Dies sieht man an der Verschiebung in x-Richtung. Das zweite was man sehen kann, ist die im vorigen Kapitel erwähnte Totzeit der beiden Trigger. Diese fällt

## 6. Strahlzeit

unterschiedlich stark aus, aufgrund der verschieden breiten NIM-Pulse, in die das Triggersignal umgewandelt wird. Da Trigger1 eine NIM-Puls-Breite von 55 ns besitzt und Trigger2 eine von 25 ns, ist die Totzeit bei Trigger1 größer. Deshalb krümmt sich die blaue Kurve deutlich stärker. Die Daten aus den Graphen sind in der unten stehenden Tabelle eingetragen.

1000 Volt Versorgungsspannung, high Gain			
Rate T1 /kHz	Rate T2 /kHz	$\mu_{rel}$	$\sigma_{gem}$
0 ± 0	0 ± 0	0 ± 0.03	54
504.00 ± 0.71	522.25 ± 0.83	33575 ± 3.03	5249
1010.78 ± 0.92	1061.89 ± 1.29	70290 ± 4.54	7860
1479.75 ± 1.09	1575.50 ± 1.41	106712 ± 5.74	9944
1932.89 ± 2.64	2084.89 ± 1.59	144168 ± 6.81	11800
2425.60 ± 2.37	2652.10 ± 2.43	187456 ± 7.96	13790
2916.00 ± 2.21	3265.67 ± 2.67	233567 ± 9.09	15740
3376.38 ± 3.35	3798.75 ± 2.77	279189 ± 10.12	17520
3793.38 ± 2.78	4321.00 ± 4.72	323287 ± 11.22	19440
4287.75 ± 4.89	4854.25 ± 3.38	369606 ± 12.12	20990
4552.25 ± 2.05	5307.25 ± 3.53	408965 ± 12.96	22450
5027.83 ± 4.63	5951.00 ± 4.43	468471 ± 14.17	24540
5398.70 ± 2.37	6463.00 ± 4.84	517400 ± 15.15	26240
5894.00 ± 3.87	7172.40 ± 3.67	590118 ± 16.62	28790
6286.43 ± 4.14	7749.71 ± 7.13	648088 ± 17.63	30540
6680.75 ± 4.47	8344.00 ± 7.68	708698 ± 18.66	32320
7075.13 ± 3.06	8957.75 ± 4.92	778209 ± 20.21	35000
7461.88 ± 3.10	9580.67 ± 5.94	856058 ± 21.51	37250
7841.67 ± 1.89	10205.33 ± 4.16	935020 ± 22.77	39440
8220.70 ± 3.00	10853.20 ± 5.84	1024523 ± 24.54	42500

In der Tabelle ist zu sehen, dass die Raten extrem stabil waren. Im unstabilsten Fall ist der relative Fehler kleiner als 0.2%. Dies ist im Gegensatz zu den geschätzten 10% der letzten Strahlzeit eine deutliche Verbesserung. Der Unterschied in den Raten wird vor allem bei großen Raten sehr deutlich. Dies hat auch wieder mit der unterschiedlichen Totzeit zu tun, welche gleich noch genauer erklärt wird.

Die Breiten können nun mit den aus der letzten Strahlzeit verglichen werden. Schaut man sich bei ähnlichen Raten das gemessene Sigma an, ist dieses bei dieser Strahlzeit im Schnitt etwa 25.5% kleiner als bei der vorigen Strahlzeit.

Man kann nun schon einmal versuchen das Schrotrauschen zu berechnen, und dieses zu entfalten. Da man bei der Berechnung jedoch auf die gemessene Rate der Trigger angewiesen ist, wird das Schrotrauschen für höhere Raten nicht richtig bestimmt. Die Berechnung des Rauschens erfolgt mit den gleichen Formeln und Methoden, die im letzten Kapitel beschrieben wurden.

In der nachfolgenden Tabelle sind die Ergebnisse dessen aufgetragen.

1000 Volt Versorgungsspannung, high Gain				
$\sigma_{gem}$	$\sigma_{shot}$ T1	$\sigma_{shot}$ T2	$\sigma_{rest}$ T1	$\sigma_{rest}$ T2
54	0.00	0.00	54.00	54.00
5249	5287.57	5194.36	637.45	755.42
7860	7488.05	7515.72	2389.29	2300.77
9944	9060.13	9260.41	4098.43	3623.24
11800	10354.86	10763.62	5658.35	4835.76
13790	11599.79	12273.64	7457.14	6286.64
15740	12718.45	13700.28	9273.00	7749.18
17520	13685.67	14978.66	10938.59	9087.91
19440	14506.20	16118.25	12941.55	10868.10
20990	15422.52	17234.29	14238.19	11981.63
22450	15891.08	18128.71	15857.99	13242.07
24540	16700.55	19402.83	17980.63	15024.70
26240	17305.54	20390.93	19724.50	16515.07
28790	18081.96	21776.76	22403.28	18831.81
30540	18674.22	22821.32	24165.37	20294.80
32320	19250.99	23864.61	25961.16	21795.93
35000	19811.06	25007.59	28853.46	24487.14
37250	20345.32	26228.62	31203.05	26450.37
39440	20856.66	27411.59	33474.07	28356.98
42500	21354.77	28693.57	36745.39	31351.70

Wie schon im vorigen Kapitel sieht man nach dem Entfalten ein deutlich kleineres verbleibendes Sigma. Allerdings verbleibt auch hier vor allem bei höheren Raten ein großes restliches Sigma. Um zu sehen, ob der Grund dafür allein die Totzeit[b][h] ist, oder noch andere Effekte auftreten, wird eine Totzeitkorrektur durchgeführt. Diese Korrektur ist nur eine Abschätzung. Sie erfolgt an folgender Formel:

$$N \approx \frac{N_m}{1 - N_m \tau / T} \quad (31)$$

Hier ist  $\tau$  die Totzeit,  $N_m$  die Anzahl der gemessenen Ereignisse,  $T$  das Zeitintervall in dem  $N_m$  bestimmt wurde und  $N$  die abgeschätzte eigentliche Zählrate. Da wir nur Raten messen und nicht tatsächliche Ereignisse, wird diese Formel etwas abewandelt. Aus der Rate kennt man den mittleren Abstand der Ereignisse, also weiss man, in welchem Zeitintervall man ein Ereignis messen müsste ( $N_m = 1$ ,  $T = 1/\text{Rate}$ ). Die Totzeit ist die genannte NIM-Puls Breite, also 25 und 55 ns. Man weiss nun, wie viele Ereignisse man statt einem einzigen Ereignis hätte messen müssen. Diesen Faktor kann nun mit der gemessenen Rate multipliziert und erhält somit die totzeitkorrigierte Version der Rate.

In Abbildung 36 ist das Ergebnis dieser Korrektur zu sehen. Wie man sieht, entsprechen beide Kurven schon besser einem linearem Zusammenhang.

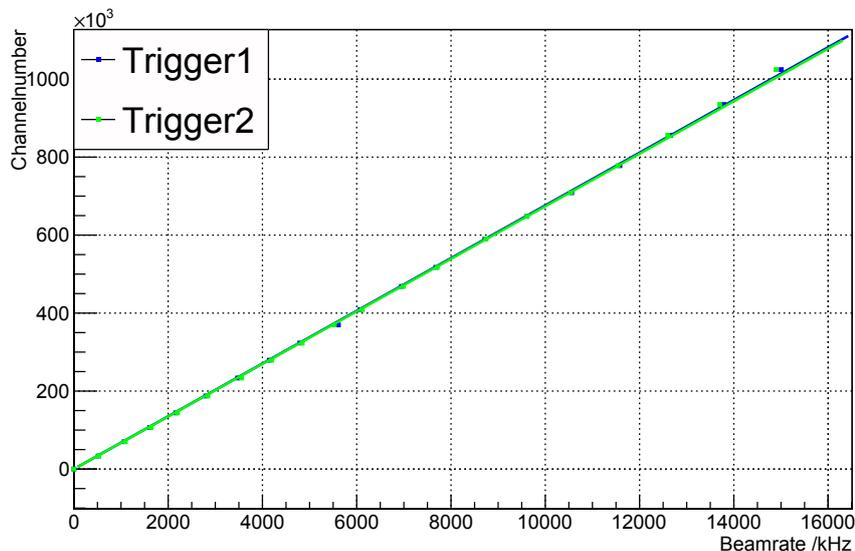


Abbildung 36: Kanalnummer in Abhängigkeit der Triggerrate. Trigger1 ist in blau, Trigger2 in grün. Diesmal wurden beide Raten auf die Totzeit korrigiert und ein linearer Fit durchgeführt.

Durch die Totzeitkorrektur und die große Skala erscheinen die Kurven nun fast Deckungsgleich. Schaut man sich die Ergebnisse des Fits an, so erkennt man, dass dies tatsächlich nicht der Fall ist, und die beiden Kurven innerhalb der Fehlerbetrachtung eine unterschiedliche Steigung aufweisen. Man sieht aber auch, dass das  $\chi^2$  einen sehr großen Wert hat, was darauf hin deutet, dass die Fehler der Messpunkte und damit auch die Fehler der Fitparameter zu klein bestimmt wurden.

$$f(\nu) = p_0 + p_1 \cdot \nu \quad (32)$$

Datensatz	Parameter		
	$p_0$	$p_1$	$\chi^2/\text{ndf}$
Trigger1	$-0.001797 \pm 0.03116$	$67.61 \pm 0.01$	8198 / 18
Trigger2	$-0.002084 \pm 0.03118$	$67.39 \pm 0.01$	$1.332 \cdot 10^4 / 18$

Um die Linearität beurteilen zu können, werden wieder die Fitgeraden von den Daten abgezogen und so der Residuen-Plot erstellt. Dieser ist in Abbildung 37 zu sehen. Die Residuen der Fits verhalten sich sehr ähnlich. Bei kleinen Raten ist der Fit zu hoch, bei großen zu klein. Vor allem bei den hohen Raten liegt der Fit sehr weit neben den gemessenen Daten. Ein Grund dafür könnte sein, dass trotz der Totzeitkorrektur die Rate noch immer zu klein bestimmt wird. Wäre dem so, könnte dies auch erklären, warum die Diskrepanz zwischen dem gesamten gemessenen Sigma und dem Schrotrauschen für höhere Raten deutlich zunimmt, wie später noch in der Tabelle ersichtlich wird.

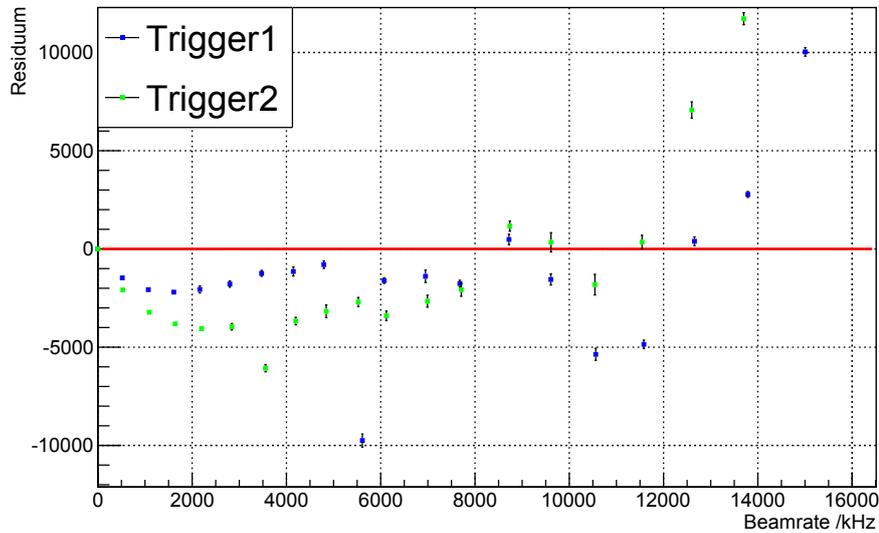


Abbildung 37: Residuen-Plot der Triggerrate. Beide Trigger zeigen ein ähnliches Verhalten. Bei niedrigen Raten ist der Fit größer als die gemessenen Daten, bei großen Raten dann zu klein.

Um die Graphen quantitativ auszuwerten, sind in der Tabelle die Messdaten aufgetragen.

1000 Volt Versorgungsspannung, Totzeit korrigiert, high Gain			
Rate T1 /kHz	Rate T2 /kHz	$\mu_{rel}$	$\sigma_{gem}$
$0 \pm 0$	$0 \pm 0$	$0 \pm 0.03$	54
$518.37 \pm 0.71$	$529.16 \pm 0.83$	$33575 \pm 3.03$	5249
$1070.28 \pm 0.92$	$1090.85 \pm 1.29$	$70290 \pm 4.54$	7860
$1610.85 \pm 1.09$	$1640.10 \pm 1.41$	$106712 \pm 5.74$	9944
$2162.82 \pm 2.64$	$2199.53 \pm 1.59$	$144168 \pm 6.81$	11800
$2799.01 \pm 2.37$	$2840.43 \pm 2.43$	$187456 \pm 7.96$	13790
$3473.00 \pm 2.21$	$3555.99 \pm 2.67$	$233567 \pm 9.09$	15740
$4146.36 \pm 3.35$	$4197.37 \pm 2.77$	$279189 \pm 10.12$	17520
$4793.47 \pm 2.78$	$4844.31 \pm 4.72$	$323287 \pm 11.22$	19440
$5610.96 \pm 4.89$	$5524.71 \pm 3.38$	$369606 \pm 12.12$	20990
$6072.69 \pm 2.05$	$6119.15 \pm 3.53$	$408965 \pm 12.96$	22450
$6949.61 \pm 4.63$	$6991.10 \pm 4.43$	$468471 \pm 14.17$	24540
$7678.74 \pm 2.37$	$7708.50 \pm 4.84$	$517400 \pm 15.15$	26240
$8721.13 \pm 3.87$	$8739.48 \pm 3.67$	$590118 \pm 16.63$	28790
$9608.66 \pm 4.14$	$9611.96 \pm 7.13$	$648088 \pm 17.63$	30540
$10561.47 \pm 4.47$	$10543.34 \pm 7.68$	$708698 \pm 18.66$	32320
$11582.10 \pm 3.06$	$11542.66 \pm 4.92$	$778209 \pm 20.21$	35000
$12655.91 \pm 3.10$	$12598.13 \pm 5.94$	$856058 \pm 21.51$	37250
$13788.57 \pm 1.89$	$13700.88 \pm 4.16$	$935020 \pm 22.77$	39440
$15005.07 \pm 3.00$	$14894.53 \pm 5.84$	$1024523 \pm 24.54$	42500

## 6. Strahlzeit

Man sieht, sowohl in der Grafik als auch in der Tabelle, dass die Raten; die von dem Plastikszintillatoren kommen, nun in etwa gleich hoch sind (wenn auch nicht gleich innerhalb der Fehlerbetrachtung). Die Diskrepanz zwischen beiden ist auch bei hohen Raten klein. Man sieht auch, dass bei sehr viel höheren Raten gemessen wurde als es vorher ersichtlich war. Die Raten von Trigger1 sind nun fast doppelt so hoch für die letzten Datenpunkte. Der Tabelle kann man auch entnehmen, dass Trigger1 viel stärker korrigiert wird, als Trigger2. Dies liegt an der mehr als doppelt so großen Totzeit von Trigger1.

Nun entfalten wir diese Ergebnisse wieder mit dem Schrotrauschen, welches von der gemessenen Rate abhängt. In der nachfolgenden Tabelle sind wieder die Ergebnisse aufgetragen.

1000 Volt Versorgungsspannung, Totzeit korrigiert, high Gain				
$\sigma_{gem}$	$\sigma_{shot}$ T1	$\sigma_{shot}$ T2	$\sigma_{rest}$ T1	$\sigma_{rest}$ T2
54	0	0	54	54
5249	5213.76	5160.34	607.17	960.69
7860	7596.26	7524.30	2019.02	2272.55
9944	9400.28	9316.08	3243.14	3477.61
11800	10960.09	10868.22	4372.24	4595.84
13790	12527.14	12435.48	5764.96	5960.12
15740	14012.45	13847.98	7169.30	7482.05
17520	15329.20	15235.78	8483.27	8649.93
19440	16508.90	16422.05	10264.97	10403.35
20990	17445.18	17580.83	11672.44	11467.11
22450	18554.55	18483.99	12638.48	12741.45
24540	19868.15	19809.11	14403.75	14484.85
26240	20875.49	20835.14	15898.17	15951.00
28790	22341.25	22317.79	18158.54	18187.37
30540	23375.32	23371.31	19654.16	19658.93
32320	24381.14	24402.09	21216.56	21192.46
35000	25565.70	25609.34	23903.87	23857.11
37250	26903.67	26965.29	25763.44	25698.94
39440	28152.46	28242.40	27621.60	27529.62
42500	29570.43	29679.96	30526.04	30419.57

Die Größe  $\sigma_{rest}$  ist auch hier deutlich kleiner als das gemessene  $\sigma$ . Außer für die zweitkleinste Rate von Trigger2 ist das verbleibende Sigma immer kleiner als es ohne die Totzeit Korrektur war. Allerdings ist bei den größten Raten das verbleibende Sigma immer noch sehr hoch. Das bedeutet, dass die Breite in der Verteilung nicht nur auf das Schrotrauschen zurück zu führen ist.

Da die Raten hier stabil waren, muss der Grund in etwas anderem liegen. Ein Grund könnte der Vorverstärker sein, welcher unter Last anfängt zu rauschen. Auch, dass die Totzeit nicht stark genug korrigiert wurde, könnte ein Grund sein. Ein sehr wahrscheinlicher Grund ist allerdings, dass der Spannungsteiler bei dieser hohen Verstärkung und den großen Raten seine Grenzen erreicht hat. Aus diesem Grund ist es so wichtig, dass man im späteren Experiment eine deutlich kleinere Verstärkung erzielen kann durch das Umschalten, da dort die Raten nochmal um einen Faktor 100-1000 größer sind. Man kann sich dies an einem Rechenbeispiel klarmachen. Trifft

## 6. Strahlzeit

den Quarzstab eine Strahlrate von etwa 1 GHz (durchschnittliche Rate im P2-Experiment pro Quarz), so erzeugt jedes Elektron etwa 100 Photonen, von denen etwa die Hälfte dem PMT erreichen. Diese werden nun vom PMT verstärkt mit einem Faktor von etwa  $10^5$  (angenommen Q.E.=1). Dann fließt von der Anode ein Strom von etwa 7.5 mA. Dies ist ein sehr hoher Strom für einen PMT. Schaut man in das Datenblatt des PMTs, ist er sogar so groß, dass er den maximal zulässigen Anodenstrom um den Faktor 50 überschreitet.

Bei unseren Tests floss aus der Anode ein Strom von etwa  $2 \mu\text{A}$ . Der PMT ist also hier noch nicht am Ende. Der Spannungsteiler selbst verbrauchte aber laut Anzeige schon mehrere hundert  $\mu\text{A}$ . Also wurden auf der kleinen Platine schon relativ große Leistungen verbraucht. Dadurch erhöht sich das thermische Rauschen.

Um diese Problematik zu vermeiden und den PMT nicht zu zerstören, wird der PMT so wie in Kapitel 2 beschrieben, bei hohen Raten mit einer kleineren Verstärkung betrieben. Aus diesem Grund und um die Verstärkungsänderung am Strahl zu testen, wurden deshalb weitere Messungen mit eingeschalteten Relais gemacht (Low Gain Mode). Wie schon in Kapitel 5.1 beschrieben muss dabei auch die Versorgungsspannung geändert werden. Dies wurde hier nicht ganz eingehalten, es wurden nun Statt dem Paar 1200 Volt High Gain - 981 Volt Low Gain, das Paar 1200 Volt High Gain - 1000 Volt Low Gain miteinander verglichen. Der Unterschied sollte also sehr gering ausfallen. Zuvor schaut man sich aber noch das gemessene Sigma bei eingeschalteten Relais an.

1000 Volt Versorgungsspannung, Totzeit korrigiert, low Gain			
Rate T1 /kHz	Rate T2 /kHz	$\mu_{rel}$	$\sigma_{gem}$
$0 \pm 0$	$0 \pm 0$	$0 \pm 0.03$	44
$3473.98 \pm 4.14$	$3518.10 \pm 2.92$	$48984 \pm 1.87$	3239
$7888.22 \pm 3.06$	$7916.63 \pm 6.52$	$118604 \pm 3.45$	5979
$13765.89 \pm 1.89$	$13671.91 \pm 6.87$	$209872 \pm 5.24$	9076

Wie man in der Tabelle sieht, ist Sigma nun deutlich kleiner als bisher. Dies liegt zum einen einfach an der kleineren Verstärkung des Spannungsteilers (sodass sich die Breite absolut verringert) und zum anderen daran, dass weniger Strom aus der Anode auch ein kleineres Schrotrauschen bedeutet.

Als letztes steht nun noch der Vergleich der Verstärkungsänderung mit den Testdaten an. Es wurde bei drei verschiedenen Raten jeweils mit 1200 Volt und 1000 Volt Betriebsspannung gemessen. Als Ratenangabe wird Trigger2 als Referenz genommen, da dort die Totzeitkorrektur nicht so groß ausfiel. Die tatsächliche Rate ist an dieser Stelle auch nicht so wichtig, sondern nur, dass sowohl im High als auch im Low Gain Mode in etwa die gleiche Rate auf den Quarz trifft, damit die Ergebnisse miteinander verglichen werden können.

Modus	$U_V / \text{V}$	Rate /kHz	$\mu_{rel}$
High Gain	1200V	$3520.73 \pm 2.31$	$839475 \pm 33.23$
Low Gain	1000V	$3518.10 \pm 2.92$	$48984 \pm 1.87$
High Gain	1200V	$7913.01 \pm 5.93$	$1850573 \pm 56.81$
Low Gain	1000V	$7916.63 \pm 6.52$	$118604 \pm 3.45$
High Gain	1200V	$13686.43 \pm 3.29$	$2768787 \pm 101.79$
Low Gain	1000V	$13671.91 \pm 6.87$	$209872 \pm 5.24$

## 6. Strahlzeit

---

Die zusammengehörigen Paare stehen hier immer direkt untereinander. Nun teilt man einfach das relative  $\mu$  durcheinander und erhält so den Verstärkungsänderung.

Rate /kHz	Verstaerkungsaenderung
$3520.73 \pm 2.31$	$17.13774 \pm 0.00094$
$7913.01 \pm 5.93$	$15.60296 \pm 0.00014$
$13686.43 \pm 3.29$	$13.19274 \pm 0.00066$

Auffällig ist direkt, dass die Änderung in der Verstärkung für höhere Raten absinkt. Auch wenn man dies mit dem Wert aus Kapitel 5.1 vergleicht, welcher bei  $22.551 \pm 0.009$  liegt, so sind diese Werte alle etwas niedriger. Da hier ein komplett anderes Setup als im Labortest verwendet wurde ist dieses Ergebnis nicht verwunderlich. Zum einen wurden keine Einzelmessungen mithilfe des CAEN ADCs digitalisiert, sondern viele Pulse mithilfe von Vorverstärkern zu einer Gleichspannung aufintegriert und mit dem QWEAK-ADC aufgezeichnet. Desweiteren wurde hier mit einem Elektronenstrahl und aus einem Quarzstab erzeugten Photonen gearbeitet und nicht mit einer LED die direkt den PMT bescheint. Man sieht jedoch auch hier, dass die Verstärkungsänderung prinzipiell funktioniert, auch wenn diese etwas geringer auszufallen scheint, als in den Labortests.

### 7. Fazit und Ausblick

Im Rahmen dieser Masterarbeit wurden erste Erfahrung mit der Messelektronik für das P2-Experiment gemacht. Dabei wurde ein eigener Spannungsteiler entwickelt, welcher die Verstärkung eines Photomultipliers mittels Relais auf der Spannungsteilerplatine umschalten kann, ohne dabei die Versorgungsspannung zu ändern. Die größte Herausforderung bestand darin, trotz umschalten der Verstärkung immer noch Gleichspannung auszukoppeln, sodass nicht nur Pulse sondern auch ein dauerhafter Strom aus der Anode fließen kann, ohne dabei die Anode auf einem undefinierten Potential zu betreiben. Dies kann bei falscher Handhabung dazu führen, dass die Anode sich erst auflädt und dann beim Anschluss an den ADC sofort entlädt und den PMT und vielleicht auch den ADC-Eingang zerstört. Die Verstärkungsänderung selbst fiel so groß aus wie erwartet (etwa Faktor 22-36), ist allerdings noch nicht so groß wie sie später einmal im P2-Experiment sein sollte. Dort ist eher ein Faktor zwischen 100 und 1000 angestrebt. Um dies zu erreichen, müssen entweder weitere Dynoden kurzgeschlossen werden, die Versorgungsspannung geändert werden oder beide Möglichkeiten kombiniert werden. Dabei gilt es noch genauer zu untersuchen, ob sich durch die Änderung der Hochspannung eventuelle Timing-Probleme in der Röhre ergeben, welche dann Probleme beim Umschalten zwischen Tracking und normaler Messung im Experiment bereiten könnten. Der aktive Spannungsteiler scheint auch bei Raten größer 10 MHz noch zu funktionieren, selbst bei hoher Verstärkung. Es könnte lediglich sein, dass dort langsam die Grenze der Leistungsfähigkeit des Spannungsteilers erreicht ist, wie in Kapitel 6.2 bei den höchsten Raten zu sehen war.

Desweiteren wurde ein vom QWEAK-Experiment entwickelter ADC in unsere eigene Datenerfassung miteingebunden. Hierfür war ein großer Programmieraufwand notwendig. Teile des Sourcecodes aus der aktuellen Datenerfassung befinden sich im Anhang. Aufgrund seiner extrem hohen Auflösung (18 Bit), die man über die Anzahl der aufaddierten Samples noch weiter steigern kann, scheint es gut für das P2-Experiment geeignet zu sein. Daneben fallen die Integralen Nichtlinearitäten sehr klein über den gesamten Dynamischen Bereich aus. Dies wurde in Kapitel 5.3 ausgewertet. Auch das Eigenrauschen des ADCs fiel extrem klein aus, sodass dies alleine keine Behinderung für die Messung von  $\sin^2\Theta_w$  mit einer Unsicherheit kleiner  $4.4 \cdot 10^{-10}$  darstellt. Weiterhin zu messen sind nun die differentiellen Nichtlinearitäten.

Die gesamte hier verwendete Elektronik konnte dann in zwei Teststrahlzeiten geprüft werden. So konnte man den PMT mit dem Spannungsteiler und dem neuen ADC direkt an einem Elektronenstrahl testen. Der Strahl selbst war leider nicht polarisiert, weswegen keine richtigen Helizitätsfenster zur Verfügung standen, allerdings konnten so schon erste Erfahrungen mit der Breite des Signals trotz sehr guten Strahlbedingungen gemacht werden. Das Schrotrauschen, welches dabei eine große Rolle spielt, sollte weiter untersucht werden. Aufgrund der begrenzten Zeit einer solchen Strahlzeit konnten nicht alle Tests mit der Ausführlichkeit getestet werden, welche man gerne angestrebt hätte. So wäre es sicher sinnvoll die Verstärkungsänderung, die während der Labortests anders ausfiel, weitere Male während einer Strahlzeit zu untersuchen.

Einmal mit dem CAEN-ADC und einer direkten Messung der Pulsladung und einmal mit dem integrierenden ADC von QWEAK bei anderen Versorgungsspannungen, aber gleichen Raten. Auf diese Weise könnte man einen direkten Vergleich zu den im Labor gemachten Messungen mit dem CAEN-ADC ziehen. Weiterhin ist die Ursache des verbleibenden Rauschens, welches sich nicht durch das Schrotrauschen erklären lässt und für höhere Raten immer größer wird, genauer zu untersuchen.

## 7. Fazit und Ausblick

---

Das gesamte P2-Experiment befindet sich noch in der Planung. Es voraussichtlich im Jahre 2019 mit der Datennahme beginnen. Ein generelles Design steht jedoch schon und ist in Abbildung 38 als kleiner Ausblick zu sehen.

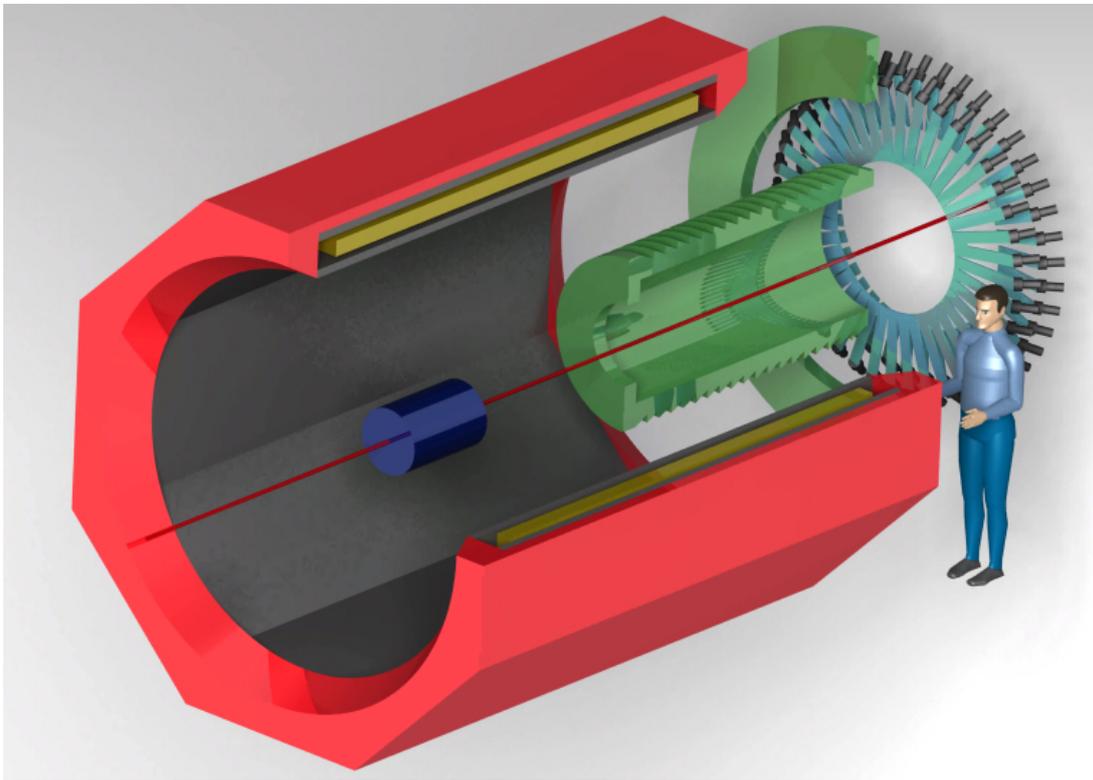


Abbildung 38: Design des P2-Experiments. Zu sehen in rot das große Joch des Elektromagneten um Elektronen falscher Energie auszusondern. In blau das flüssige Wasserstofftarget in dem der Strahl mit den Atomkernen stößt. In hellblau und schwarz der Ring aus PMTs um das Signal aufzufangen. In grün die Abschirmungen aus Blei um die PMTs und Quarze vor Photonen und anderer Sekundärstrahlung zu schützen.



## A. Anhang

### A.1. Programmcode

#### T\_QWEAK.cxx

```
1  #include "TVMEModule.hxx"
2  #include "T_QWEAK.hxx"
3  #include "vmebus.h"
4
5  T_QWEAK::T_QWEAK()
6  {
7      pointer = NULL;
8  }
9  T_QWEAK::~T_QWEAK()
10 {
11     delete pointer;
12 }
13 struct memory_struct_QWEAK* T_QWEAK::MemoryMapModule(off_t padd, size_t size)
14 {
15     if (padd==0)
16         padd = baseaddress;
17     pointer = (memory_struct_QWEAK *) vmestd(padd, sizeof (memory_struct_QWEAK));
18     return pointer;
19 }
20 int T_QWEAK::GetEventCounter()
21 {
22     return eventcounter;
23 }
24 struct memory_struct_QWEAK* T_QWEAK::GetPointer()
25 {
26     return pointer;
27 }
```

## T\_QWEAK.hxx

```

1  #ifndef __T_QWEAK__
2  #define __T_QWEAK__
3
4  #include <stdio.h>
5
6  #ifdef __cplusplus
7  #include "TVMEModule.hxx"
8  #endif
9
10 struct QWEAK_channel_struct {
11     /* Each ADC channel has a total size of 0x18 bytes.
12     * The internal structure of the channel bank is
13     * as follows.
14     *
15     * Register name  Offset  Description
16     *-----
17     * 04 - Block 1 sum
18     * 08 - Block 2 sum
19     * 0C - Block 3 sum
20     * 10 - Block 4 sum
21     * 14 - Total sum
22     * 18 - Acquisition period number
23     */
24     volatile unsigned long block[4]; /* 00 - Block 1 sum */
25     /* 04 - Block 2 sum */
26     /* 08 - Block 3 sum */
27     /* 0C - Block 4 sum */
28     volatile unsigned long total; /* 10 - Total sum */
29     volatile unsigned long nsamples_seqnum; /* 14 - Acquisition period number */
30 };
31
32 struct memory_struct_QWEAK {
33     /* register name  Offset  Description
34     *-----
35     * 18 - ADC2 data
36     * 30 - ADC3 data
37     * 48 - ADC4 data
38     * 60 - ADC5 data
39     * 78 - ADC6 data
40     * 90 - ADC7 data
41     * A8 - ADC8 data
42     *
43     * C0 - Clock cycles per sample
44     * equals sample_period+40
45     * C1 - Number of blocks per
46     * gate: 1-4, default is 4
47     * C2 - Samples per block
48     *
49     * C4 - Number of clock cycles
50     * delay for first sample
51     * per gate
52     * C5 - Selector for the gate
53     * and clock source
54     * C6 - Gate frequency equals
55     * 100kHz/int_gate_freq
56     *
57     * C8 - IRQ control register
58     * C9 - Interrupt vector
59     * CA - Data ready per channel
60     * CB - Firmware Revision Number
61     *
62     * CC - Firmware Revision Date
63     *
64     * D0 - FIFO and Sequence number reset
65     */
66     volatile struct QWEAK_channel_struct adc_channels[8]; /* 00 - ADC1 data */
67     /* 18 - ADC2 data */
68     /* 30 - ADC3 data */
69     /* 48 - ADC4 data */
70     /* 60 - ADC5 data */
71     /* 78 - ADC6 data */
72     /* 90 - ADC7 data */
73     /* A8 - ADC8 data */
74     volatile unsigned long c0_c1_c2; /* C0 - Clock cycles per sample */
75     /* equals sample_period+40 */
76     /* C1 - Number of blocks per */
77     /* gate: 1-4, default is 4 */
78     /* C2 - Samples per block */
79     volatile unsigned long c4_c5_c6; /* C4 - Number of clock cycles */
80     /* delay for first sample */
81     /* per gate */
82     /* C5 - Selector for the gate */
83     /* and clock source */
84     /* C6 - Gate frequency equals */
85     /* 100kHz/int_gate_freq */
86     volatile unsigned long c8_c9_ca_cb; /* C8 - IRQ control register */
87     /* C9 - Interrupt vector */
88     /* CA - Data ready per channel */
89     /* CB - Firmware Revision Number */
90     volatile unsigned long firmware_rev_date; /* CC - Firmware Revision Date */
91     volatile unsigned long reset_fifo_seqnum; /* D0 - FIFO and Sequence number reset */
92 };
93
94 #ifdef __cplusplus
95 #include "common.h"
96
97 /* Define Masks */
98 const unsigned long QWEAK_period_mask = 0xff;
99 const unsigned long QWEAK_nblocks_mask = 0x3;
100 const unsigned long QWEAK_blocknsamples_mask = 0xCFFF;
101
102 const unsigned long QWEAK_seqnum_mask[2] = {0x00FF0000, 0x0000FF00}; // Mask for sequence number
103 const unsigned long QWEAK_nsamples_mask[2] = {0x0000FFFF, 0xFFFF0000}; // Mask for number of samples
104 const unsigned long QWEAK_seqnum_shift[2] = {16, 8}; // Bit shift for sequence number
105 const unsigned long QWEAK_nsamples_shift[2] = {0, 16}; // Bit shift for number of samples
106
107 /* Access functions */
108 void QWEAK_SetVerboseLevel(unsigned int verb);
109 void QWEAK_SetDefaultSamplePeriod(unsigned short sampleperiod);
110 void QWEAK_SetDefaultNumberOFBlocks(unsigned short numblocks);
111 void QWEAK_SetDefaultSamplesPerBlock(unsigned int samp_per_block);
112 void QWEAK_SetDefaultGateDelay(unsigned short gatedelay);
113 void QWEAK_SetDefaultGateClockSources(unsigned short gateflag, unsigned short clkflag);
114 void QWEAK_SetDefaultIntGateFreq(unsigned int intgatefreq);
115
116 void QWEAK_SetForwardsReading();
117 void QWEAK_SetBackwardsReading();
118
119 int QWEAK_Init (volatile struct memory_struct_QWEAK *ADC);
120 void QWEAK_PrintFirmwareVersion(volatile struct memory_struct_QWEAK *ADC);
121 void QWEAK_PrintStatus(volatile struct memory_struct_QWEAK *ADC);
122
123 void QWEAK_SetSamplePeriod(volatile struct memory_struct_QWEAK *ADC, unsigned short sampleperiod);
124 void QWEAK_SetNumberOFBlocks(volatile struct memory_struct_QWEAK *ADC, int numblocks);
125 void QWEAK_SetSamplesPerBlock(volatile struct memory_struct_QWEAK *ADC, unsigned int samp_per_block);
126 void QWEAK_SetGateDelay(volatile struct memory_struct_QWEAK *ADC, unsigned short gatedelay);
127 void QWEAK_SetGateClockSources(volatile struct memory_struct_QWEAK *ADC, unsigned short gateflag, unsigned short clkflag);
128 void QWEAK_SetGateClockSourcesVerbose(volatile struct memory_struct_QWEAK *ADC, unsigned short gateflag, unsigned short clkflag, int verbose);
129 void QWEAK_SetIntGateFreq(volatile struct memory_struct_QWEAK *ADC, unsigned int intgatefreq);
130
131 unsigned short QWEAK_ProbeDataReady(volatile struct memory_struct_QWEAK *ADC);
132
133 unsigned long QWEAK_WaitForDataReady(volatile struct memory_struct_QWEAK *ADC, unsigned short rmask, long timeout);
134
135 int QWEAK_ReadRegisters(volatile struct memory_struct_QWEAK *ADC, unsigned int channel);
136 int QWEAK_ReadForwards(volatile struct memory_struct_QWEAK *ADC, unsigned int channel);
137
138 int QWEAK_Read(volatile struct memory_struct_QWEAK *ADC, unsigned short rmask, unsigned int channel, long timeout, long verbose);
139
140 void QWEAK_ClearAllChannels(volatile struct memory_struct_QWEAK *ADC);
141
142 /* Define global variables */
143 int kReadBackwards = 0; /* Set to 1 for backward reading */
144 int kVerboseLevel = 2; /* Set to 0 for only errors,
145 * set to 1 for short status messages,
146 * and 2 for full messages. */

```

## A. Anhang

```
120 int QWEAK = 0;
121 //struct QWEAK_struct missingadc;          /* An empty ADC structure */
122
123 /* Default values to be used during initialization */
124 unsigned short kDefaultPeriod = 0;        /* 40 cycles per sample */
125 unsigned short kDefaultBlocks = 1;        /* 4 blocks per gate */
126 unsigned int kDefaultSamples = 500;       /* 500 samples per block */
127 unsigned short kDefaultDelay = 20;        /* Wait 20 samples after gate */
128 unsigned short kDefaultGateFlg = 1;       /* External gate */
129 unsigned short kDefaultClkFlg = 0;        /* Internal clock */
130 unsigned int kDefaultGateFreq = 3300;     /* Gate period is 33.00 ms */
131
132
133
134 int QWEAK_get_short(unsigned long word, int position){
135 // Grab a specific unsigned short out of a unsigned long
136 if (position>=0 && position<=3){
137 return ((word >> (position*8))&0xFF);
138 } else {
139 // This is a bad position.
140 return 0;
141 }
142 };
143
144 int QWEAK_get_int(unsigned long word, int position){
145 // Grab a specific unsigned int out of a unsigned long
146 if (position>=0 && position<=1){
147 return ((word >> (position*16))&0xFFFF);
148 } else {
149 // This is a bad position.
150 return 0;
151 }
152 };
153
154 unsigned long QWEAK_set_short(unsigned long word, int position, int newdata){
155 // Set a specific unsigned short within a unsigned long
156 int offset;
157 unsigned long mask, invmask;
158 if (position>=0 && position<=3){
159 offset = position*8;
160 mask = 0xFF << offset;
161 invmask = 0xFFFFFFF ^ mask;
162 return ( (word&invmask) + ((newdata<<offset)&mask) );
163 } else {
164 // This is a bad position.
165 return word;
166 }
167 };
168
169 unsigned long QWEAK_set_int(unsigned long word, int position, int newdata){
170 // Set a specific unsigned int within a unsigned long
171 int offset;
172 unsigned long mask, invmask;
173 if (position>=0 && position<=1){
174 offset = position*16;
175 mask = 0xFFFF << offset;
176 invmask = 0xFFFFFFF ^ mask;
177 return ( (word&invmask) + ((newdata<<offset)&mask) );
178 } else {
179 // This is a bad position.
180 return word;
181 }
182 };
183
184
185 void QWEAK_SetVerboseLevel(unsigned int verb){
186 if (verb>0) kVerboseLevel=verb;
187 else kVerboseLevel=0;
188 };
189
190 void QWEAK_SetDefaultSamplePeriod(unsigned short sampleperiod){
191 kDefaultPeriod = sampleperiod;
192 };
193 void QWEAK_SetDefaultNumberOfBlocks(unsigned short numblocks){
194 kDefaultBlocks = numblocks;
195 };
196 void QWEAK_SetDefaultSamplesPerBlock(unsigned int samp_per_block){
197 kDefaultSamples = samp_per_block;
198 };
199 void QWEAK_SetDefaultGateDelay(unsigned short gatedelay){
200 kDefaultDelay = gatedelay;
201 };
202 void QWEAK_SetDefaultGateClockSources(unsigned short gateflag, unsigned short clkflag){
203 kDefaultGateFlg = gateflag;
204 kDefaultClkFlg = clkflag;
205 };
206 void QWEAK_SetDefaultIntGateFreq(unsigned int intgatefreq){
207 kDefaultGateFreq = intgatefreq;
208 };
209
210 void QWEAK_SetForwardsReading(){kReadBackwards=0;};
211 void QWEAK_SetBackwardsReading(){kReadBackwards=1;};
212
213 /*****
214 *
215 * QWEAKInit - Initialize Qweak ADC library. Disable/Clear Counters
216 *
217 *
218 * RETURNS: 0;
219 */
220
221 int QWEAK_Init (volatile struct memory_struct_QWEAK *ADC){
222
223 QWEAK_SetSamplePeriod(ADC, kDefaultPeriod);
224 QWEAK_SetNumberOfBlocks(ADC, kDefaultBlocks);
225 QWEAK_SetSamplesPerBlock(ADC, kDefaultSamples);
226 QWEAK_SetGateDelay(ADC, kDefaultDelay);
227 QWEAK_SetIntGateFreq(ADC, kDefaultGateFreq);
228 QWEAK_SetGateClockSources(ADC, kDefaultGateFlg,kDefaultClkFlg);
229
230
231 /* QWEAKPrintStatus(ii); */
232 printf("Finished Initializing QWEAK BOARD \n");
233 return 0;
234 }
235
236 /*****
237 *
238 * QWEAK_PrintStatus - Print the status of the ADC to the console.
239 */
```

## A. Anhang

```
240 * RETURNS: nothing
241 */
242
243 void QWEAK_PrintStatus(volatile struct memory_struct_QWEAK *ADC){
244     unsigned long ldata1, ldata2, ldata3;
245
246     ldata1 = ADC->c0_c1_c2;
247     ldata2 = ADC->c4_c5_c6;
248     ldata3 = ADC->c8_c9_ca_cb;
249
250
251
252     printf("Variable dump:\n\tclk \tbnblk \tmsamp\tdelay\tgateclk\tfreq\ti_cr\ti_vec\tready\trevnum\n\t0x%02x \t0x%02x \t0x%04x \t0x%02x \t0x%02x \t0x%04x \t
253     QWEAK_get_short(ldata1,3), QWEAK_get_short(ldata1,2), QWEAK_get_int(ldata1,0),
254     QWEAK_get_short(ldata2,3), QWEAK_get_short(ldata2,2), QWEAK_get_int(ldata2,0),
255     QWEAK_get_short(ldata3,3), QWEAK_get_short(ldata3,2), QWEAK_get_short(ldata3,1), QWEAK_get_short(ldata3,0)
256     );
257
258 }
259
260
261 /*****
262 *
263 * QWEAK_SetSamplePeriod - Set the ADC to ...
264 *
265 * RETURNS: nothing
266 */
267 void QWEAK_SetSamplePeriod(volatile struct memory_struct_QWEAK *ADC, unsigned short sampleperiod)
268 {
269     unsigned long ldata1;
270     ldata1 = ADC->c0_c1_c2;
271     ADC->c0_c1_c2 = QWEAK_set_short(ldata1,3,sampleperiod);
272     ldata1 = ADC->c0_c1_c2;
273 }
274
275
276
277 /*****
278 *
279 * QWEAK_SetNumberOfBlocks - Set the ADC to ...
280 *
281 * RETURNS: nothing
282 */
283 void QWEAK_SetNumberOfBlocks(volatile struct memory_struct_QWEAK *ADC, int numblocks)
284 {
285     unsigned long ldata1;
286     if (numblocks<0 || numblocks>4){
287         printf("\tThe number of blocks must be between 1 and 4.\n");
288     } else {
289         /* This is a good ADC structure. */
290         ldata1 = ADC->c0_c1_c2;
291         ADC->c0_c1_c2 = QWEAK_set_short(ldata1,2,(numblocks-1));
292         ldata1 = ADC->c0_c1_c2;
293     }
294 }
295
296
297 /*****
298 *
299 * QWEAK_SetSamplesPerBlock - Set the ADC to ...
300 *
301 * RETURNS: nothing
302 */
303 void QWEAK_SetSamplesPerBlock(volatile struct memory_struct_QWEAK *ADC, unsigned int samp_per_blk)
304 {
305     unsigned long ldata1;
306     if (samp_per_blk==0 || samp_per_blk>16383){
307         printf("\tThe number of samples per block should be between 1 and 16383.\n");
308     } else {
309         /* This is a good ADC structure. */
310         ldata1 = ADC->c0_c1_c2;
311         ADC->c0_c1_c2 = QWEAK_set_int(ldata1,0,samp_per_blk);
312         ldata1 = ADC->c0_c1_c2;
313     }
314 }
315
316
317
318
319 /*****
320 *
321 * QWEAK_SetGateDelay - Set the ADC to ...
322 *
323 * RETURNS: nothing
324 */
325 void QWEAK_SetGateDelay(volatile struct memory_struct_QWEAK *ADC, unsigned short gatedelay)
326 {
327     unsigned long ldata2;
328     if (gatedelay==0){
329         printf("\tThe gate delay should be at least 1 cycle.\n");
330     } else {
331         /* This is a good ADC structure. */
332         ldata2 = ADC->c4_c5_c6;
333         ADC->c4_c5_c6 = QWEAK_set_short(ldata2,3,gatedelay);
334         ldata2 = ADC->c4_c5_c6;
335     }
336 }
337
338
339
340 /*****
341 *
342 * QWEAK_SetGateClockSources - Set the ADC to use the internal or external
343 * sources for the GATE and the SYSTEM_CLOCK.
344 *
345 * RETURNS: nothing
346 */
347 void QWEAK_SetGateClockSources(volatile struct memory_struct_QWEAK *ADC, unsigned short gateflag, unsigned short clkflag)
348 {
349     QWEAK_SetGateClockSourcesVerbose(ADC, gateflag, clkflag, 0);
350 };
351
352 void QWEAK_SetGateClockSourcesVerbose(volatile struct memory_struct_QWEAK *ADC, unsigned short gateflag,
353     unsigned short clkflag, int verbose)
354 {
355     unsigned long ldata2;
356     unsigned short source = 0x00;
357     float freq;
358     if (gateflag !=0)    source |= 0x02;
359 }
```

## A. Anhang

```
360 if (clkflag !=0) source |= 0x01;
361 /* This is a good ADC structure. */
362 ldata2 = ADC->c4_c5_c6;
363 ADC->c4_c5_c6 = QWEAK_set_short(ldata2,2,source);
364 if (verbose>0){
365     ldata2 = ADC->c4_c5_c6;
366     source = QWEAK_get_short(ldata2,2);
367     if ((source & 0x1) == 0){
368         printf("\tUsing INTERNAL 20 Mhz CLOCK.\n");
369     } else {
370         printf("\tUsing EXTERNAL CLOCK.\n");
371     }
372     if ((source & 0x2) == 0){
373         printf("\tUsing INTERNAL GATE.\n");
374     } else {
375         printf("\tUsing EXTERNAL GATE.\n");
376     }
377 }
378
379 }
380
381 /*****
382 *
383 * QWEAK_SetIntGateFreq - Set the ADC to ...
384 *
385 * RETURNS: nothing
386 */
387 void QWEAK_SetIntGateFreq(volatile struct memory_struct_QWEAK *ADC, unsigned int intgatefreq)
388 {
389     unsigned long ldata2;
390     float freq;
391     ldata2 = ADC->c4_c5_c6;
392     ADC->c4_c5_c6 = QWEAK_set_int(ldata2,0,intgatefreq);
393     ldata2 = ADC->c4_c5_c6;
394     freq = 100.0/(intgatefreq);
395 }
396
397
398 void QWEAK_ClearAllChannels(volatile struct memory_struct_QWEAK *ADC){
399     int ii, jj;
400     unsigned long tmpval;
401
402     ADC->reset_fifo_seqnum = 0xFFFFFFFF;
403
404     for(jj=0;jj<4;jj++){
405         for(ii=0;ii<8;ii++){
406             tmpval = ADC->adc_channels[ii].nsamples_seqnum;
407         }
408     }
409 }
410
411 /*****
412 *
413 * QWEAK_PrintFirmwareVersion - Prints the Firmware Version...
414 *
415 * RETURNS: nothing
416 */
417 void QWEAK_PrintFirmwareVersion(volatile struct memory_struct_QWEAK *ADC){
418     static char* months[12] = {"JAN", "FEB", "MAR", "APR", "MAY", "JUN",
419                               "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"};
420     unsigned long laddr;
421     unsigned long versioncode, datecode;
422     unsigned int version, day, mon, year;
423
424     /* This is a good ADC structure. */
425     laddr = (unsigned long)ADC;
426     versioncode = QWEAK_get_short(ADC->c8_c9_ca_cb,0); // firmware_revision;
427     version = ((versioncode>>4)&0xF)*10 + ((versioncode)&0xF);
428     datecode = ADC->firmware_rev_date;
429     day = ((datecode>>28) & 0xF)*10
430         + ((datecode>>24) & 0xF);
431     year = ((datecode>>12) & 0xF)*1000
432         + ((datecode>>8) & 0xF)*100
433         + ((datecode>>4) & 0xF)*10
434         + ((datecode) & 0xF);
435     mon = ((datecode>>16) & 0xFF) - 0x41; // The numeric month with JAN=0
436     printf("QWEAK ADC, address=0x%lx\n",laddr);
437     printf("\tFirmware revision ID: %d (raw code=0x%02x)\n",version,
438           versioncode);
439     printf("\tFirmware revision date: %02d / %s / %d (raw code=0x%08x)\n",
440           day, months[mon], year, datecode);
441 }
442
443 unsigned long QWEAK_WaitForDataReady(volatile struct memory_struct_QWEAK *ADC,unsigned short rmask, long timeout){
444     long cycles;
445     short dataready;
446
447     cycles = -1;
448     if (ADC != NULL){
449         cycles = 0;
450
451         if (timeout>0) {
452             while (cycles<timeout && dataready!=rmask) {
453                 dataready = (QWEAK_ProbeDataReady(ADC)) & rmask;
454                 cycles++;
455             }
456             if (cycles >= timeout){
457                 cycles *= -1;
458             }
459         } else {
460             while (cycles<timeout && dataready!=rmask) {
461                 dataready = (QWEAK_ProbeDataReady(ADC)) & rmask;
462                 cycles++;
463             }
464         }
465     }
466     return cycles;
467 }
468
469 int QWEAK_ReadRegisters(volatile struct memory_struct_QWEAK *ADC, unsigned int channel){
470     int status;
471
472     status = QWEAK_ReadForwards(ADC, channel);
473 }
474
475
476
477
478
479
```

## A. Anhang

---

```
480     return status;
481 };
482
483 int QWEAK_ReadForwards(volatile struct memory_struct_QWEAK *ADC, unsigned int channel){
484     int ii, jj;
485     int nvalues;
486     unsigned long tmpval;
487     nvalues = 0;
488
489     return ADC->adc_channels[channel].nsamples_seqnum;
490 };
491
492
493
494
495 int QWEAK_Read(volatile struct memory_struct_QWEAK *ADC, unsigned short rmask, unsigned int channel, long timeout, long verbose){
496     int ii, jj;
497     int ready;
498     int nvalues;
499     unsigned long tmpval;
500
501     if (ADC!=NULL){
502         ready = QWEAK_WaitForDataReady(ADC, rmask, timeout);
503         if (ready < 0){
504             if (verbose == 1)
505                 printf("No data was available after %d cycles.\n",-1*ready);
506             } else {
507                 nvalues = QWEAK_ReadRegisters(ADC, channel);
508             }
509         }
510     return nvalues;
511 };
512
513 unsigned short QWEAK_ProbeDataReady(volatile struct memory_struct_QWEAK *ADC)
514 {
515     short dataflag = 0;
516     if (ADC!=NULL){
517         dataflag = QWEAK_get_short(ADC->c8_c9_ca_cb,1);
518     }
519     return dataflag;
520 }
521
522
523 #endif
524
525 #ifdef __cplusplus
526 class T_QWEAK : public TVMEModule
527 {
528 public:
529     T_QWEAK();
530     ~T_QWEAK();
531     struct memory_struct_QWEAK* MemoryMapModule(off_t padd=0, size_t size=sizeof(struct memory_struct_QWEAK));
532     int GetEventCounter();
533     struct memory_struct_QWEAK* GetPointer();
534 protected:
535 private:
536     unsigned long eventcounter;
537     struct memory_struct_QWEAK *pointer;
538 };
539 #endif
540
541 #endif
```

## DAQcore.h

```

1  #ifndef __DAQCORE__
2  #define __DAQCORE__
3
4  #include "T_CAENV792.hxx"
5  #include "T_CAENV830.hxx"
6  #include "T_QWEAK.hxx"
7
8  #include <sys/time.h>
9  #include <sys/types.h>
10 #include <sys/ipc.h>
11 #include <sys/msg.h>
12 #include <signal.h>
13
14 // bitmap for process states:
15 #define PSsuccess 0
16 #define PSfailed 1
17 #define PSpreparing 2
18 #define PSstarting 4
19 #define PSrunning 8
20 #define PSstopping 16
21 #define PSfinished 32
22 #define PSinterrupted 64
23
24 #define ADCexists(x) ( (x>=1)&&(x<=2) )
25
26 // 'extern "C"' is needed in order to link this C and the C++ code together later:
27 #ifdef __cplusplus
28 extern "C" {
29 #endif
30
31 struct DAQconfig
32 {
33     int mypid;
34     int msgid;
35     int *status;
36     char *mode;
37     long sched_runtime;
38     int polainvert;
39     volatile struct memory_struct_CAENV792 *ADC1, *ADC2;
40     volatile struct memory_struct_CAENV830 *scaler1, *scaler2;
41     volatile struct memory_struct_QWEAK *QDC;
42
43     int Pipe[2]; // pipes for communication between server and DAQ child process
44     fd_set PipeState; // a file descriptor set
45     char outfileName1[512], outfileName2[512];
46     char errmsg[512];
47 };
48
49 int DAQcore(struct DAQconfig *cfg);
50 // common functions to be used by DAQ routines:
51 void ResetADC();
52 void Reset2ADCs();
53
54 void Finish();
55
56 ////////////////////////////////////////////////////
57 // specialized DAQ routines: //
58 int DAQ_test();
59 int DAQ_ADCChannel0();
60 int DAQ_ADCChannelSelect();
61 int DAQ_ADCChannelSelectHisto();
62 int DAQ_QWEAKChannelSelectHisto();
63 // specialized exit (write data to disk and cleanup) routines: //
64 int EXIT_test(int signal);
65 int EXIT_standard();
66 ////////////////////////////////////////////////////
67
68 int InitADC(volatile struct memory_struct_CAENV792 *ADC, int which);
69 int InitQWEAK_ADC(volatile struct memory_struct_QWEAK *ADC, int which);
70
71 void WriteDataToFile(int maxDetectors, int maxADCchannels, int maxPolas, FILE **outfile, unsigned long *data);
72
73 int WritePipe(int pipe, char *xmit);
74 int WritePipe_ERROR(int pipe, char *xmit);
75 void ListenSignal(struct sigaction *act, void* function, int signal);
76 void ListenSIGNALSignal();
77 void ListenEXITSignal();
78 void IgnoreSIGNALSignal();
79 void IgnoreEXITSignal();
80 int CreateMsgQueue(key_t key);
81 int ConnectMsgQueue(key_t key);
82
83 // #ifdef __cplusplus
84 #ifndef __cplusplus
85
86 int success;
87 char *successstring[2] = {"failed", "success"};
88
89 //void *data;
90 //FILE *outfile1, *outfile2;
91
92 struct DAQmode
93 {
94     const char *command;
95     int (*function)();
96     int (*signalfunction)();
97     int (*exitfunction)();
98     //const char *format;
99 };
100
101 struct DAQconfig *config = NULL;
102 int whichDAQmode = -1;
103 // DAQ_control is a global variable meant to signal a DAQ_xxx function, that it should do
104 // something special or that an event occurred:
105 int DAQcontrol = 0;
106
107 struct sigaction EXITact, SIGNALact;
108 #define EXIT_signal SIGUSR1
109 #define SIGNAL_signal SIGUSR2
110
111 // This is a list possible DAQ modes. The first entry is the "command" or name of the
112 // DAQ mode, the second entry is the function to be called to acquire the data,
113 // and the third field hold the function to be called in order to write the data to
114 // disk and exit.

```

## A. Anhang

---

```
115 // Note, that some of the functions can easily be "recycled".
116 //
117 // N.B.: DO NOT CHANGE THE SYNTAX OF THE FOLLOWING "struct":
118 // ===== If you need to introduce any changes here, first refer to the source code of
119 // function "GetDAQmodes()" in daccord_server.cxx!!
120 // Even additional blanks between the curved brackets and the DAQ mode names can
121 // lead to strange behaviour!
122 // A maximum of 64 DAQ modes with a maximum length of the name of 29 characters
123 // each is allowed!
124 //
125 struct DAQmode DAQmodeList[] =
126 {
127     /* test mode: to check communication, signal processing etc. */
128     {"test", DAQ_test, NULL, EXIT_test},
129     /* basic modes: histogram single adc, pol/unpol */
130     {"ADCChannel0", DAQ_ADCChannel0, NULL, EXIT_standard},
131     /* basic modes: histogram of some channels of the adc */
132     {"ADCChannelSelect", DAQ_ADCChannelSelect, NULL, EXIT_standard},
133     /* basic modes: histogram of some channels of the adc with a light coding */
134     {"ADCChannelSelectHisto", DAQ_ADCChannelSelectHisto, NULL, EXIT_standard},
135     /* basic modes: histogram of some channels of the QWEAK_adc */
136     {"QWEAKChannelSelectHisto", DAQ_QWEAKChannelSelectHisto, NULL, EXIT_standard},
137     {NULL, NULL, NULL, NULL}
138 };
139 //endif
140 #endif
141 #ifndef __cplusplus
142 }
143 #endif
144 #endif
145 #endif
146 #endif
147
```

## DAQcore.c

```

1  #include "DAQcore.h"
2
3  #include <stdio.h>
4  #include <unistd.h>
5  #include <stdlib.h>
6  #include <string.h>
7  #include <errno.h>
8  #include <time.h>
9
10 int Fxx_TO_ADC2ch[32] = { -1,
11                          0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12                          16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
13                          -1, -1, -1, -1, -1, -1, -1, -1 };
14
15 int ADC2ch_TO_Fxx[32] = {
16     1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
17     -1, -1, -1, -1,
18     13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
19     -1, -1, -1, -1
20 };
21
22 /* Values that can be changed in the QWEAK_ADC_setting.dat file, here: explanation only */
23 //unsigned short QWEAK_Period = x; /* clock period(internal 20MHz) * (x+40) cycles per sample (minimum 2us/sample) */
24 //unsigned short QWEAK_Blocks = x; /* x blocks per gate */
25 //unsigned int QWEAK_Samples = x; /* x samples per block (use even numbers only) */
26 //unsigned short QWEAK_Delay = x; /* wait x samples after gate +2.5us (at least 1 is recommended) */
27 //unsigned int QWEAK_GateFreq = x; /* Gate period is x/100 ms or for Frequency its 100/x kHz */
28 //unsigned short QWEAK_GateFlg = 0,1; /* 0 Internal, 1 External gate */
29 //unsigned short QWEAK_ClkFlg = 0,1; /* 0 Internal, 1 external clock */
30
31 char pathname[512] = "/data/kpha4/rda2/P2_testbeam_27_05_2015"; //pathname for data, also change the "basepath" in /AnalysisGUI/analysiswindow.cpp
32 int DAQcore(struct DAQconfig *cfg)
33 //void *DAQcore(void *voidconfig)
34 {
35     int i;
36     //struct memory_struct_CAENV792 *poi = NULL;//config->QDC;
37     success = 1;
38     config = cfg;
39     // report "starting" to server:
40     WritePipe(config->Pipe[1], "starting");
41     // get own pid and try to open MsgQueue for IPC:
42     config->mypid = getpid();
43     //printf("***** my pid = %d *****\n", config->mypid);
44     for (i=0; i<10; i++)
45     {
46         config->msgid = ConnectMsgQueue((key_t)config->mypid);
47         if (config->msgid==-1) break;
48         usleep(1000);
49     }
50     if (config->msgid==-1)
51     printf("WARNING - Cannot open MsgQueue for IPC with daccord_server: %s\n", strerror(errno));
52     for (i=0; DAQmodeList[i].command!=NULL; i++)
53     {
54         success = 0; // we're not yet successful finding the command string!
55         if (!strcmp(config->mode, DAQmodeList[i].command))
56         {
57             success=1; // success!!
58             whichDAQmode = i;
59             printf("DAQcore: starting DAQ mode %s...\n", DAQmodeList[i].command);
60             DAQmodeList[i].function();
61             break;
62         }
63     };
64     msgctl(config->msgid, IPC_RMID, NULL);
65     if (success==0)
66     printf("DAQcore: DAQ has failed...\n");
67     printf("DAQcore: exiting now...\n");
68     WritePipe(config->Pipe[1], "finished");
69     return 0;
70 }
71
72 ////////////////////////////////////////////////////////////////////////////////////////
73 ////////////////////////////////////////////////////////////////////////////////////////
74 ////////////////////////////////////////////////////////////////////////////////////////
75 ////////////////////////////////////////////////////////////////////////////////////////
76 ////////////////////////////////////////////////////////////////////////////////////////
77 ////////////////////////////////////////////////////////////////////////////////////////
78 ////////////////////////////////////////////////////////////////////////////////////////
79 ////////////////////////////////////////////////////////////////////////////////////////
80 ////////////////////////////////////////////////////////////////////////////////////////
81 ////////////////////////////////////////////////////////////////////////////////////////
82 ////////////////////////////////////////////////////////////////////////////////////////
83 ////////////////////////////////////////////////////////////////////////////////////////
84 ////////////////////////////////////////////////////////////////////////////////////////
85 ////////////////////////////////////////////////////////////////////////////////////////
86 ////////////////////////////////////////////////////////////////////////////////////////
87 ////////////////////////////////////////////////////////////////////////////////////////
88 ////////////////////////////////////////////////////////////////////////////////////////
89 ////////////////////////////////////////////////////////////////////////////////////////
90 ////////////////////////////////////////////////////////////////////////////////////////
91 ////////////////////////////////////////////////////////////////////////////////////////
92 ////////////////////////////////////////////////////////////////////////////////////////
93 ////////////////////////////////////////////////////////////////////////////////////////
94 ////////////////////////////////////////////////////////////////////////////////////////
95 ////////////////////////////////////////////////////////////////////////////////////////
96 ////////////////////////////////////////////////////////////////////////////////////////
97 ////////////////////////////////////////////////////////////////////////////////////////
98 ////////////////////////////////////////////////////////////////////////////////////////
99 ////////////////////////////////////////////////////////////////////////////////////////
100 ////////////////////////////////////////////////////////////////////////////////////////
101 ////////////////////////////////////////////////////////////////////////////////////////
102 ////////////////////////////////////////////////////////////////////////////////////////
103 ////////////////////////////////////////////////////////////////////////////////////////
104 ////////////////////////////////////////////////////////////////////////////////////////
105 ////////////////////////////////////////////////////////////////////////////////////////
106 ////////////////////////////////////////////////////////////////////////////////////////
107 ////////////////////////////////////////////////////////////////////////////////////////
108 ////////////////////////////////////////////////////////////////////////////////////////
109 ////////////////////////////////////////////////////////////////////////////////////////
110 ////////////////////////////////////////////////////////////////////////////////////////
111 ////////////////////////////////////////////////////////////////////////////////////////
112 ////////////////////////////////////////////////////////////////////////////////////////
113 ////////////////////////////////////////////////////////////////////////////////////////
114 ////////////////////////////////////////////////////////////////////////////////////////
115 ////////////////////////////////////////////////////////////////////////////////////////
116 ////////////////////////////////////////////////////////////////////////////////////////
117 ////////////////////////////////////////////////////////////////////////////////////////
118 ////////////////////////////////////////////////////////////////////////////////////////
119 ////////////////////////////////////////////////////////////////////////////////////////
120
121 int DAQ_test()
122 {
123     ListenEXITSignal();
124     printf("DAQ_test: switch to 'running' status\n");
125     WritePipe(config->Pipe[1], "running\n");
126     for (;;)
127     {
128         printf("."); fflush(stdout);
129         usleep(100000);
130     }
131     return 0;
132 }
133
134 ////////////////////////////////////////////////////////////////////////////////////////
135 ////////////////////////////////////////////////////////////////////////////////////////
136 ////////////////////////////////////////////////////////////////////////////////////////
137 ////////////////////////////////////////////////////////////////////////////////////////
138 ////////////////////////////////////////////////////////////////////////////////////////
139 ////////////////////////////////////////////////////////////////////////////////////////
140 ////////////////////////////////////////////////////////////////////////////////////////
141 ////////////////////////////////////////////////////////////////////////////////////////
142 ////////////////////////////////////////////////////////////////////////////////////////
143 ////////////////////////////////////////////////////////////////////////////////////////
144 ////////////////////////////////////////////////////////////////////////////////////////
145 ////////////////////////////////////////////////////////////////////////////////////////
146 ////////////////////////////////////////////////////////////////////////////////////////
147 ////////////////////////////////////////////////////////////////////////////////////////
148 ////////////////////////////////////////////////////////////////////////////////////////
149 ////////////////////////////////////////////////////////////////////////////////////////
150 ////////////////////////////////////////////////////////////////////////////////////////
151 ////////////////////////////////////////////////////////////////////////////////////////
152 ////////////////////////////////////////////////////////////////////////////////////////
153 ////////////////////////////////////////////////////////////////////////////////////////
154 ////////////////////////////////////////////////////////////////////////////////////////
155 ////////////////////////////////////////////////////////////////////////////////////////
156 ////////////////////////////////////////////////////////////////////////////////////////
157 ////////////////////////////////////////////////////////////////////////////////////////
158 ////////////////////////////////////////////////////////////////////////////////////////
159 ////////////////////////////////////////////////////////////////////////////////////////
160 ////////////////////////////////////////////////////////////////////////////////////////
161 ////////////////////////////////////////////////////////////////////////////////////////
162 ////////////////////////////////////////////////////////////////////////////////////////
163 ////////////////////////////////////////////////////////////////////////////////////////
164 ////////////////////////////////////////////////////////////////////////////////////////
165 ////////////////////////////////////////////////////////////////////////////////////////
166 ////////////////////////////////////////////////////////////////////////////////////////
167 ////////////////////////////////////////////////////////////////////////////////////////
168 ////////////////////////////////////////////////////////////////////////////////////////
169 ////////////////////////////////////////////////////////////////////////////////////////
170 ////////////////////////////////////////////////////////////////////////////////////////
171 ////////////////////////////////////////////////////////////////////////////////////////
172 ////////////////////////////////////////////////////////////////////////////////////////
173 ////////////////////////////////////////////////////////////////////////////////////////
174 ////////////////////////////////////////////////////////////////////////////////////////
175 ////////////////////////////////////////////////////////////////////////////////////////
176 ////////////////////////////////////////////////////////////////////////////////////////
177 ////////////////////////////////////////////////////////////////////////////////////////
178 ////////////////////////////////////////////////////////////////////////////////////////
179 ////////////////////////////////////////////////////////////////////////////////////////
180 ////////////////////////////////////////////////////////////////////////////////////////
181 ////////////////////////////////////////////////////////////////////////////////////////
182 ////////////////////////////////////////////////////////////////////////////////////////
183 ////////////////////////////////////////////////////////////////////////////////////////
184 ////////////////////////////////////////////////////////////////////////////////////////
185 ////////////////////////////////////////////////////////////////////////////////////////
186 ////////////////////////////////////////////////////////////////////////////////////////
187 ////////////////////////////////////////////////////////////////////////////////////////
188 ////////////////////////////////////////////////////////////////////////////////////////
189 ////////////////////////////////////////////////////////////////////////////////////////
190 ////////////////////////////////////////////////////////////////////////////////////////
191 ////////////////////////////////////////////////////////////////////////////////////////
192 ////////////////////////////////////////////////////////////////////////////////////////
193 ////////////////////////////////////////////////////////////////////////////////////////
194 ////////////////////////////////////////////////////////////////////////////////////////
195 ////////////////////////////////////////////////////////////////////////////////////////
196 ////////////////////////////////////////////////////////////////////////////////////////
197 ////////////////////////////////////////////////////////////////////////////////////////
198 ////////////////////////////////////////////////////////////////////////////////////////
199 ////////////////////////////////////////////////////////////////////////////////////////
200 ////////////////////////////////////////////////////////////////////////////////////////
201 ////////////////////////////////////////////////////////////////////////////////////////
202 ////////////////////////////////////////////////////////////////////////////////////////
203 ////////////////////////////////////////////////////////////////////////////////////////
204 ////////////////////////////////////////////////////////////////////////////////////////
205 ////////////////////////////////////////////////////////////////////////////////////////
206 ////////////////////////////////////////////////////////////////////////////////////////
207 ////////////////////////////////////////////////////////////////////////////////////////
208 ////////////////////////////////////////////////////////////////////////////////////////
209 ////////////////////////////////////////////////////////////////////////////////////////
210 ////////////////////////////////////////////////////////////////////////////////////////
211 ////////////////////////////////////////////////////////////////////////////////////////
212 ////////////////////////////////////////////////////////////////////////////////////////
213 ////////////////////////////////////////////////////////////////////////////////////////
214 ////////////////////////////////////////////////////////////////////////////////////////
215 ////////////////////////////////////////////////////////////////////////////////////////
216 ////////////////////////////////////////////////////////////////////////////////////////
217 ////////////////////////////////////////////////////////////////////////////////////////
218 ////////////////////////////////////////////////////////////////////////////////////////
219 ////////////////////////////////////////////////////////////////////////////////////////
220
221 int DAQ_ADCChannel0()
222 {
223     int i;
224     int event=0;
225     volatile struct memory_struct_CAENV792 *ADC=NULL;
226     struct CAENV792_eventblock eventblock;
227     unsigned long chs=0;
228     FILE *outfile=NULL;
229     char *outfilename=NULL;
230     FILE *information;
231     information = fopen("information.dat", "w");
232     // get pointer to VME module:
233     ADC = config->ADC1;
234
235     // initialize and configure ADC:
236     if (InitADC(ADC, 1))
237     {
238         // initialization of ADC failed:
239         fprintf(information, "the initialization failed");
240         usleep(1000000);
241         return 0;
242     }
243     CAENV792_ResetADC(ADC);
244     fprintf(information, "The ADC is reseted\n");
245
246     // configuring the file to open
247     if (strlen(config->outfilename1)>0)
248     {
249         //create the file name
250         outfilename = malloc(strlen(pathname)+strlen("/data")+strlen(config->outfilename1)+strlen(".dat")+1);
251         sprintf(outfilename, "%s/data/%s.dat", pathname, config->outfilename1);
252     }

```

## A. Anhang

```
120     printf("trying to open file: >%s<\n", outfile);
121     //create the file
122     outfile = fopen(outfile, "w");
123 }
124 else{
125     outfile = malloc(strlen(pathname)+strlen("/data/")+strlen("noName")+strlen(".dat")+1);
126     sprintf(outfile, "%s/data/noName.dat", pathname);
127     printf("trying to open file: >%s<\n", outfile);
128     //create the file
129     outfile = fopen(outfile, "w");
130 }
131 fprintf(information, "file created\n");
132
133
134 // clear eventblock:
135 CAENV792_ClearEventblock(&eventblock);
136 fprintf(information, "The eventblock is ready to be used\n");
137 ListenEXITSignal();
138
139 WritePipe(config->Pipe[1], "running");
140
141
142 ///////////////////////////////////////////////////////////////////
143 // Now we can start the measurement loop: ///////////////////////////////////////////////////////////////////
144 for (;;)
145 {
146
147     if ( (ADC->StatusRegister1) & CAENV792_DATAREADY )
148     {
149
150         chs = CAENV792_GetEvent(&eventblock, ADC);
151
152         for (i=0; i<chs; i++)
153         {
154             if (eventblock.channel[i]==0)
155             {
156                 event++;
157                 fprintf(outfile, "%ld\n", eventblock.data[i]);
158                 if(event==100){
159                     event=0;
160                     fflush(outfile);
161                 }
162             }
163         }
164
165         if (*(config->status)==PSstopping) break;
166     } ///////////////////////////////////////////////////////////////////
167     fprintf(information, "measurement are done\n");
168     fclose(outfile);
169     fclose(information);
170     return 0;
171 }
172
173
174 ///////////////////////////////////////////////////////////////////
175 ///////////////////////////////////////////////////////////////////
176 int DAQ_ADCChannelSelect() // to acquire all the 16 channel of the adc
177 {
178     int i;//used to read all channels
179     int j;//used to create all outputfiles
180     int event=0;
181     volatile struct memory_struct_CAENV792 *ADC=NULL;
182     struct CAENV792_eventblock eventblock;
183     clock_t start;
184     unsigned long chs=0;
185     FILE *outfile[16];
186     char *outfilename[16];
187     FILE *information;
188     char channelfilename[256];
189     FILE* channelfile;
190     char tmp1[64], tmp2[64], tmp3[64];
191
192     start=clock();
193
194     information = fopen("information.dat", "w");
195     // get pointer to VME module:
196     ADC = config->ADC1;
197
198     // initialize and configure ADC:
199     // here it uses the usedchannel_2 file in order to keep the 1st one clean for
200     // one channel data acquisition mode histo.
201     if (InitADC(ADC, 2))
202     {
203         // initialization of ADC failed:
204         fprintf(information, "the initialization failed");
205         usleep(1000000);
206         return 0;
207     };
208     CAENV792_ResetADC(ADC);
209     fprintf(information, "The ADC is reseted\n");
210
211     //read the file that knows which file to create
212     sprintf(channelfilename, "usedchannels_adc2.dat");
213     channelfile = fopen(channelfilename, "r");
214     // configuring the file to open
215     if (strlen(config->outfilename1)>0)
216     {
217         //create the files names
218         for(j=0; j<16; j++){
219             fscanf(channelfile, "%s %s %s \n", tmp1, tmp2, tmp3);
220             if(strcmp("yes", tmp3)==0){
221                 outfilename[j] = malloc(strlen(pathname)+strlen("/data/ch00/") +strlen(config->outfilename1)+strlen("_ch00")+strlen(".dat")+1);
222                 if(j<10){
223                     sprintf(outfilename[j], "%s/data/ch0%d/%s_ch0%d.dat", pathname, j, config->outfilename1, j);
224                     printf("trying to open file: >%s<\n", outfilename[j]);
225                 }else{
226                     sprintf(outfilename[j], "%s/data/ch%d/%s_ch%d.dat", pathname, j, config->outfilename1, j);
227                     printf("trying to open file: >%s<\n", outfilename[j]);
228                 }
229                 //create the file
230                 outfile[j] = fopen(outfilename[j], "w");
231                 if(outfile[j]==NULL)
232                     { perror("couldn't create data file");
233                       return 0;
234                     }
235                 fprintf(outfile[j], "#EventMode\n");
236             }
237         }
238     }
239     else
240     {
241         outfile[j]=NULL;
242     }
243 }
```

## A. Anhang

```
240     outfile[j]=NULL;
241 }
242 fscanf(channelfile,"%s %s %s\n", tmp1, tmp2, tmp3);
243 }
244 }
245 else{ //only used by the web interface
246 for(j=0;j<16;j++){
247     outfile[j] = malloc(strlen(pathname)+strlen("/data/ch00/") +strlen("noName")+strlen("_ch00")+strlen(".dat")+1);
248     if(j<10){
249         sprintf(outfile[j], "%s/data/ch0%d/noName_ch0%d.dat",pathname,j,j);
250         printf("trying to open file: >%s<\n", outfile[j]);
251     }else{
252         sprintf(outfile[j], "%s/data/ch%d/noName_ch%d.dat",pathname,j,j);
253         printf("trying to open file: >%s<\n", outfile[j]);
254     }
255     //create the file
256     outfile[j] = fopen(outfile[j], "w");
257 }
258 }
259 fclose(channelfile);
260 fprintf(information,"file created\n");
261 }
262 // clear eventblock:
263 CAENV792_ClearEventblock(&eventblock);
264 fprintf(information,"The eventblock is ready to be used\n");
265 ListenEXITSsignal();
266
267 WritePipe(config->Pipe[1], "running");
268
269
270 ///////////////////////////////////////////////////////////////////
271 // Now we can start the measurement loop: ///////////////////////////////////////////////////////////////////
272 for (;;)
273 {
274     {
275         if ( (ADC->StatusRegister1) & CAENV792_DATAREADY )
276         {
277             chs = CAENV792_GetEvent(&eventblock, ADC);
278             for (i=0; i<chs; i++)
279             {
280                 //eventblock.channel[i] is even (channel number * 2) due to the 2 different types of reading an input (we are reading low resolution)
281                 fprintf(outfile[eventblock.channel[i]/2], "%ld\n", eventblock.data[i]);
282             }
283             //allow to update the file frequently
284             if((clock()-start)>event*100){
285                 event++;
286                 for (i=0; i<chs; i++)
287                 {
288                     fflush(outfile[eventblock.channel[i]/2]);
289                 }
290             }
291         }
292     }
293     if (*(config->status)==PSstopping) break;
294 }
295 ///////////////////////////////////////////////////////////////////
296 ///////////////////////////////////////////////////////////////////
297 fprintf(information,"measurement are done\n");
298 for(i=0;i<16;i++){fclose(outfile[i]);}
299 fclose(information);
300 return 0;
301 }
302
303 ///////////////////////////////////////////////////////////////////
304 ///////////////////////////////////////////////////////////////////
305 int DAQ_ADCChannelSelectHisto() // to acquire all the 16 channel of the adc
306 {
307     int i;//used to read all channels
308     int j;//used to create all outputfiles
309     int event=0;
310     int DAQChannels_events[16][4096];
311     clock_t start=0;
312     volatile struct memory_struct_CAENV792 *ADC=NULL;
313     struct CAENV792_eventblock eventblock;
314     unsigned long chs=0;
315     FILE *outfile[16];
316     char *outfilename[16];
317     FILE *information;
318     char channelfilename[256];
319     FILE* channelfile;
320     char tmp1[64], tmp2[64], tmp3[64];
321
322     printf("Now entered %s\n",__PRETTY_FUNCTION__);
323
324     //initialize all the arrays
325     for(i=0;i<16;i++)
326     {
327         memset(DAQChannels_events[i],0,4096*sizeof(int));
328     }
329
330     //set the timer beginning
331     start=clock();
332
333     information = fopen("information.dat", "w");
334     // get pointer to VME module:
335     ADC = config->ADC1;
336
337     // initialize and configure ADC:
338     // here it uses the usedchannel_2 file in order to keep the 1st one clean for
339     // one channel data acquisition mode histo.
340     if (InitADC(ADC, 2))
341     {
342         // initialization of ADC failed:
343         fprintf(information,"the initialization failed");
344         printf("*** ERROR *** ADC initialization failed!\n");
345         usleep(1000000);
346         fclose(information);
347         return 0;
348     };
349     CAENV792_ResetADC(ADC);
350     fprintf(information,"The ADC is reseted\n");
351
352     //read the file that knows which file to create
353     sprintf(channelfilename, "usedchannels_adc2.dat");
354 }
```

## A. Anhang

```
360 channelfile = fopen(channelfilename, "r");
361 if (channelfile == NULL)
362 {
363     printf("Error opening channel file %s: %s\n", channelfilename, strerror(errno));
364     fclose(information);
365     return 0;
366 }
367 // configuring the file to open
368 printf("config->outfilename1 reads: %s\n", config->outfilename1);
369 if (strlen(config->outfilename1)>0)
370 {
371     //create the files names
372     for(j=0;j<16;j++){
373         fscanf(channelfile, "%s %s %s \n", tmp1, tmp2, tmp3);
374     }
375     if(strcmp("yes", tmp3)==0){
376         outfile[j] = malloc(strlen(pathname)+strlen("/data/ch00")+strlen(config->outfilename1)+strlen("_ch00")+strlen(".dat")+1);
377         if(j<10){
378             sprintf(outfile[j], "%s/data/ch0%d/%s_ch0%d.dat", pathname, j, config->outfilename1, j);
379             printf("trying to open file: %s<\n", outfile[j]);
380         }else{
381             sprintf(outfile[j], "%s/data/ch%d/%s_ch%d.dat", pathname, j, config->outfilename1, j);
382             printf("trying to open file: %s<\n", outfile[j]);
383         }
384         //create the file
385         outfile[j] = fopen(outfile[j], "w");
386         if(outfile[j]==NULL)
387             {perror("couldn't create data file");
388              fclose(information);
389              return 0;
390             }
391         fprintf(outfile[j], "#HistoChannel\n");
392         fclose(outfile[j]);
393     }
394     //not read channel
395     else{
396         outfile[j]=NULL;
397         outfile[j]=NULL;
398     }
399     fscanf(channelfile, "%s %s %s \n", tmp1, tmp2, tmp3);
400 }
401 }
402 else{ //only used by the web interface
403     for(j=0;j<16;j++){
404         outfile[j] = malloc(strlen(pathname)+strlen("/data/ch00")+strlen("noName")+strlen("_ch00")+strlen(".dat")+1);
405         if(j<10){
406             sprintf(outfile[j], "%s/data/ch0%d/noName_ch0%d.dat", pathname, j, j);
407             printf("trying to open file: %s<\n", outfile[j]);
408         }else{
409             sprintf(outfile[j], "%s/data/ch%d/noName_ch%d.dat", pathname, j, j);
410             printf("trying to open file: %s<\n", outfile[j]);
411         }
412         //create the file
413         outfile[j] = fopen(outfile[j], "w");
414     }
415 }
416 fclose(channelfile);
417 fprintf(information, "file created\n");
418
419 // clear eventblock:
420 CAENV792_ClearEventblock(&eventblock);
421 printf(information, "The eventblock is ready to be used\n");
422 ListenEXITSignal();
423
424 WritePipe(config->Pipe[1], "running");
425
426
427
428 ////////////////////////////////////////////////////
429 // Now we can start the measurement loop: ////////////////////////////////////////////////////
430 for (;;)
431 {
432     if ( (ADC->StatusRegister1) & CAENV792_DATAREADY )
433     {
434         chs = CAENV792_GetEvent(&eventblock, ADC);
435         for (i=0; i<chs; i++)
436         {
437             //eventblock.channel[i] is even (channel number * 2) due to the 2 different types of reading an input (we are reading low resolution)
438             DAQChannels_events[eventblock.channel[i]/2][eventblock.data[i]]++;
439         }
440         //allow to update the file frequently every 0.1 seconds (event counts how many writing we already did)
441         if((clock()-start)>event*100){
442             event++;
443             for (i=0; i<chs; i++)
444             {
445                 if(outfile[eventblock.channel[i]/2]!=NULL)
446                 {
447                     outfile[eventblock.channel[i]/2] = fopen(outfile[eventblock.channel[i]/2], "w");
448                     fprintf(outfile[eventblock.channel[i]/2], "#HistoMode\n");
449                     for(j=0;j<4096;j++)
450                     {
451                         fprintf(outfile[eventblock.channel[i]/2], "%d\n", DAQChannels_events[eventblock.channel[i]/2][j]);
452                     }
453                     fclose(outfile[eventblock.channel[i]/2]);
454                 }
455             }
456         }
457         if (*(config->status)==PSstopping) break;
458     }
459     ////////////////////////////////////////////////////
460     fprintf(information, "measurement are done\n");
461     for (i=0; i<chs; i++)
462     {
463         if(outfile[eventblock.channel[i]/2]!=NULL){
464             outfile[eventblock.channel[i]/2] = fopen(outfile[eventblock.channel[i]/2], "w");
465             fprintf(outfile[eventblock.channel[i]/2], "#HistoMode\n");
466             for(j=0;j<4096;j++)
467             {
468                 fprintf(outfile[eventblock.channel[i]/2], "%d\n", DAQChannels_events[eventblock.channel[i]/2][j]);
469             }
470         }
471     }
472 }
```



## A. Anhang

```
600     fscanf(channelfile, "%s %s %s \n", tmp1, tmp2, tmp3);
601
602     if(strcmp("yes", tmp3)==0){
603         outfilename[j] = malloc(strlen(pathname)+strlen("/data/ch00")+strlen(config->outfilename1)+strlen("_ch00")+strlen(".dat")+1);
604         if(j<10){
605             sprintf(outfilename[j], "%s/data/ch0%d/%s_ch0%d.dat", pathname, j, config->outfilename1, j);
606             printf("trying to open file: >%s<\n", outfilename[j]);
607         }else{
608             sprintf(outfilename[j], "%s/data/ch%d/%s_ch%d.dat", pathname, j, config->outfilename1, j);
609             printf("trying to open file: >%s<\n", outfilename[j]);
610         }
611         //create the file
612         outfile[j] = fopen(outfilename[j], "w");
613         if(outfile[j]==NULL)
614             {perror("couldn't create data file");
615              fclose(information);
616              return 0;
617             }
618         fprintf(outfile[j], "#HistoModeQWEAK\n");
619         fprintf(outfile[j], "%d\n", cols);
620         fclose(outfile[j]);
621     }
622     //not read channel
623     else{
624         outfile[j]=NULL;
625         outfilename[j]=NULL;
626         printf("channelnumber %d no outfile\n", j);
627     }
628     fscanf(channelfile, "%s %s %s \n", tmp1, tmp2, tmp3);
629 }
630 }
631 else{ //only used by the web interface
632     for(j=0;j<8;j++){
633         outfilename[j] = malloc(strlen(pathname)+strlen("/data/ch00")+strlen("noName")+strlen("_ch00")+strlen(".dat")+1);
634         if(j<10){
635             sprintf(outfilename[j], "%s/data/ch0%d/noName_ch0%d.dat", pathname, j, j);
636             printf("trying to open file: >%s<\n", outfilename[j]);
637         }else{
638             sprintf(outfilename[j], "%s/data/ch%d/noName_ch%d.dat", pathname, j, j);
639             printf("trying to open file: >%s<\n", outfilename[j]);
640         }
641         //create the file
642         outfile[j] = fopen(outfilename[j], "w");
643     }
644 }
645 fclose(channelfile);
646 fprintf(information, "file created\n");
647
648
649
650
651
652 fprintf(information, "The eventblock is ready to be used\n");
653 ListenEXITSIGNAL();
654
655 WritePipe(config->Pipe[1], "running");
656
657
658 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
659 // Now we can start the measurement loop: ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
660 for (;;)
661 {
662
663
664     if (QWEAK_ProbeDataReady(ADC)!=0 )
665     {
666         test++;
667
668
669         for (i=0; i<chs; i++)
670         {
671             if(test % 10000 == 0 && outfilename[i]!=NULL){ //just prints out which channel number every 10000th event was
672                 printf("%ld , channel %d \n", ADC->adc_channels[i].total+cols, i);
673                 printf("%ld #ADC, channel %d \n", ADC->adc_channels[i].total, i);
674                 printf("%ld #max, channel %d \n", cols, i);
675                 //printf("%ld , aquisition period number", ADC->adc_channels [i].nsamples_seqnum);
676             }
677             DAQChannels_events[i][ADC->adc_channels[i].total+cols]++;
678         }
679         QWEAK_ClearAllChannels(ADC);
680     }
681     //if(QWEAK_ProbeDataReady(ADC)!=0) QWEAK_ClearAllChannels(ADC);
682
683     if (*(config->status)==PSstopping) break;
684 } ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
685 fprintf(information, "measurement are done\n");
686 printf("\n number of measurement points: %ld \n", test);
687 printf("\n measurement are done\n");
688 for (i=0; i<chs; i++)
689 {
690     if(outfilename[i]!=NULL){
691         outfile[i] = fopen(outfilename[i], "w");
692         fprintf(outfile[i], "#HistoModeQWEAK\n");
693         fprintf(outfile[i], "%d\n", 2*cols);
694
695         for(j=0;j<2*cols;j++)
696         {
697             fprintf(outfile[i], "%d\n", DAQChannels_events[i][j]);
698         }
699         fclose(outfile[i]);
700     }
701 }
702
703 fclose(information);
704 return 0;
705 }
706
707 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
708 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
709 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
710 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
711 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
712 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
713 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
714 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
715 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
716 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
717 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
718 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
719 int EXIT_test(int signal)
```

## A. Anhang

```

720 {
721     int j;
722     printf("EXIT_test: begin\n");
723     IgnoreEXITSIGNAL(); // just to be sure...
724     usleep(100000);
725     printf("EXIT_test: switch to 'stopping' status\n");
726     WritePipe(config->Pipe[1], "stopping");
727     printf("EXIT_test: starting readout:\n");
728     for (j=0; j<256; j++)
729     {
730         printf("*"); fflush(stdout);
731         usleep(10000);
732     };
733     usleep(100000);
734     printf("EXIT_test: switch to 'finished' status\n");
735     WritePipe(config->Pipe[1], "finished");
736     printf("EXIT_test: done!\n");
737     Finish();
738     return 0;
739 }
740 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
741 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
742 int EXIT_standard()
743 {
744     printf("exite is launched");
745     IgnoreEXITSIGNAL();
746     *(config->status) = PSstopping;
747     WritePipe(config->Pipe[1], "stopping");
748     return 0;
749 }
750 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
751 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
752 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
753 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
754
755 int InitQWEAK_ADC(volatile struct memory_struct_QWEAK *ADC, int which)
756 {
757     char channelfilename[256], errmsg[512];
758     FILE* channelfile;
759
760     sprintf(channelfilename, "usedchannels_adc%d.dat", which);
761     channelfile = fopen(channelfilename, "r");
762     if(channelfile == NULL)
763     {
764         success = 0;
765         sprintf(errmsg, "DAQcore::InitADC: Cannot open file >%s<", channelfilename);
766         perror("DAQcore::InitADC");
767         WritePipe_ERROR(config->Pipe[1], errmsg);
768         usleep(100000);
769         WritePipe(config->Pipe[1], "stopping");
770         return 1;
771     }
772     else
773     {
774
775         printf("initialize ADC\n");
776         QWEAK_Init(ADC);
777         QWEAK_PrintFirmwareVersion(ADC);
778     }
779
780     return 0;
781 }
782 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
783 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
784 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
785 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
786 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
787 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
788
789 int InitADC(volatile struct memory_struct_CAENV792 *ADC, int which)
790 {
791     char channelfilename[256], errmsg[512];
792     FILE* channelfile;
793     int i;
794     char tmp1[200], tmp2[200], tmp3[200];
795
796     ADC->BitClear1 = CAENV792_SELECTADDRESS; // select address via Rotary Switch
797     ADC->BitSet2 = CAENV792_DISABLEZEROSUP; // disables zero suppression
798     ADC->BitSet2 = CAENV792_DISABLEOVERFUSUP; // disables overflow suppression
799     ADC->BitClear2 = CAENV792_HOWINCREMENTTRIG; // otherwise the counter does crazy things...
800     ADC->EventCounterReset = 0x1; // clears the event counter
801
802     sprintf(channelfilename, "usedchannels_adc%d.dat", which);
803     channelfile = fopen(channelfilename, "r");
804     if(channelfile == NULL)
805     {
806         success = 0;
807         sprintf(errmsg, "DAQcore::InitADC: Cannot open file >%s<", channelfilename);
808         perror("DAQcore::InitADC");
809         WritePipe_ERROR(config->Pipe[1], errmsg);
810         usleep(100000);
811         WritePipe(config->Pipe[1], "stopping");
812         return 1;
813     }
814     else
815     {
816         // The following stuff is highly dangerous and MUST be changed into something that makes more sense!
817         for(i=0; i<32; i++)
818         {
819             fscanf(channelfile, "%s %s %s", tmp1, tmp2, tmp3);
820             if(strcmp("no", tmp3)==0) ADC->Threshold[i] = 0x100; // set the "kill" bit for this channel
821             if(strcmp("yes", tmp3)==0) ADC->Threshold[i] = 0x000; // clear " " " " " "
822         };
823         fclose(channelfile);
824     };
825
826     return 0;
827 }
828 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
829 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
830 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
831 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
832
833 void WriteDataToFile(int maxDetectors, int maxADCchannels, int maxPolas, FILE **outFile, unsigned long *data)
834 {
835     int f, j, p;
836     for (p=0; p<maxPolas; p++)
837     for (j=0; j<maxADCchannels; j++)
838     for (f=0; f<maxDetectors; f++)
839         fprintf(outFile[p], "%ld\t", data[ f + j*maxDetectors + p*maxADCchannels ]);

```

## A. Anhang

```
840     fprintf(outfile[p], "\n");
841     };
842 }
843 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
844 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
845 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
846 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
847 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
848 void Finish()
849 {
850     WritePipe(config->Pipe[1], "finished");
851     close(config->Pipe[0]);
852     close(config->Pipe[1]);
853     config->ADC1 = NULL;
854     config->ADC2 = NULL;
855     config->scaler1 = NULL;
856     exit(0);
857 }
858 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
859 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
860 int WritePipe(int pipe, char *xmit)
861 {
862     char auxbuff[strlen(xmit)+16];
863     sprintf(auxbuff, "%s: %s\n", successstring[success], xmit);
864     return write(pipe, auxbuff, strlen(auxbuff));
865 }
866 int WritePipe_ERROR(int pipe, char *xmit)
867 {
868     char auxbuff[strlen(xmit)+16];
869     sprintf(auxbuff, "ERROR: %s\n", xmit);
870     return write(pipe, auxbuff, strlen(auxbuff));
871 }
872 void ListenSignal(struct sigaction *act, void* function, int signal)
873 {
874     act->sa_sigaction = NULL;
875     act->sa_handler = (void*)function;
876     act->sa_flags = 0;
877     sigaction(signal, act, NULL);
878 }
879 void ListenSIGNALsignal()
880 {
881     // listen to SIGNAL signal, e.g. SIGUSR2:
882     ListenSignal(&SIGNALact, (void*)DAQmodeList[whichDAQmode].signalfunction, SIGNAL_signal);
883     return;
884 }
885 void ListenEXITsignal()
886 {
887     // listen to EXIT signal, e.g. SIGUSR1:
888     ListenSignal(&EXITact, (void*)DAQmodeList[whichDAQmode].exitfunction, EXIT_signal);
889     return;
890 }
891 void IgnoreSIGNALsignal()
892 {
893     // ignore e.g. SIGUSR1:
894     SIGNALact.sa_sigaction = NULL;
895     SIGNALact.sa_handler = SIG_IGN;
896     SIGNALact.sa_flags = 0;
897     sigaction(SIGNAL_signal, &SIGNALact, NULL);
898     return;
899 }
900 void IgnoreEXITsignal()
901 {
902     // ignore e.g. SIGUSR1:
903     EXITact.sa_sigaction = NULL;
904     EXITact.sa_handler = SIG_IGN;
905     EXITact.sa_flags = 0;
906     sigaction(EXIT_signal, &EXITact, NULL);
907     return;
908 }
909 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
910 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
911 int CreateMsgQueue(key_t key)
912 {
913     return msgget(key, 0666|IPC_CREAT);
914 }
915 int ConnectMsgQueue(key_t key)
916 {
917     return msgget(key, 0666);
918 }
```

## B. Abbildungs- und Tabellenverzeichnis

### Abbildungsverzeichnis

1.	$\sin^2\Theta_w$ in Abhängigkeit des Impulsübertrags Q	2
2.	Schematische Zeichnung eines PMTs	4
3.	Schematische Zeichnung eines PMTs	6
4.	Quanteneffizienz als Funktion der Wellenlänge	7
5.	Schema eines passiven Spannungsteilers für einen PMT	11
6.	Schema eines aktiven Spannungsteilers	12
7.	Erstes und zweites Glied der Teilerkette	13
8.	Finaler Schaltplan des Spannungsteiler-Prototyps	15
9.	Layout des Spannungsteiler-Prototyps	16
10.	Foto des Gesamtaufbaus des Spannungsteiler-Prototyps	16
11.	Quantisierung eines analog Signals (grau) in digitale Werte (rote Punkte)	17
12.	Mögliche Abweichung realer von idealen ADCs	19
13.	Stufungsfehler, Breite der Stufe (links), missing codes (rechts)	20
14.	Foto der Front des CAEN Boards	23
15.	Foto des des QWEAK Boards	24
16.	Schema zum Testen der relativen Verstärkung	25
17.	1200V $U_{Versorgung}$ Relais aus (High Gain Mode)	26
18.	982V $U_{Versorgung}$ Relais ein (Low Gain Mode)	27
19.	Verstärkung im High Gain Mode	28
20.	Verstärkung im High Gain Mode, Residuen Plot	29
21.	Verstärkung im Low Gain Mode	29
22.	Verstärkung im Low Gain Mode, Residuen Plot	30
23.	Verstärkungsänderung des Spannungsteilers	32
24.	Verstärkungsänderung beider Dynoden in Abhängigkeit der Betriebsspannung	34
25.	Aufnahme des Eigenrauschens mithilfe des Kurzschließens des ADC-Eingangs	37
26.	Messergebnisse durch das Spannungsnormale	39
27.	Überprüfen der Linearität des ADCs	41
28.	Residuen Plot aus Fit mit Polynom zweiten Grads	42
29.	Residuen-Plot durch sequenzielles Fitten	43
30.	Schematischer Messaufbau der Strahlzeit	44
31.	Foto des Messaufbaus der Strahlzeit	46
32.	Ergebnisse der Strahlzeit bei 1000 Volt und 300 Sekunden Messzeit	46
33.	Messergebnisse der Strahlzeit bei 1000 Volt Versorgungsspannung	48
34.	Messergebnisse der Strahlzeit bei 1200 Volt Versorgungsspannung	49
35.	Mittlere getroffene Kanalnummer in Abhängigkeit der beiden Triggerraten	52
36.	Kanalnummer in Abhängigkeit der beiden Triggerraten, Totzeit korrigiert	55
37.	Residuen-Plot der Triggerraten	56
38.	Design des P2-Experiments	61

## C. Quellen

### Literatur

- [a] Glenn F. Knoll.(2000). *Radiation Detection and Measurement*; John Wiley & Sons
- [b] W.R. Leo.(1994). *Techniques for Nuclear and Particle Physics Experiments, A How-to Approach*; Springer-Verlag
- [c] Phillips Photonics.(1994).*Photomultiplier Tubes, Principles and Applications*; Phillips
- [d] Hamamatsu Photonics K.K. *Photomultiplier Tubes, Basics and Applications*  
[http://sales.hamamatsu.com/assets/applications/ETD/pmt\\_handbook\\_complete.pdf](http://sales.hamamatsu.com/assets/applications/ETD/pmt_handbook_complete.pdf)
- [e] T. Allison, M. Anderson, D. Androic, D.S. Armstrong, A. Asaturyan, T. Averett, R. Averill, J. Balewski, J. Beaufait, R.S. Beminiwattha, J. Benesch, F. Benmokhtar, J. Bessuille, J. Birchall, E. Bonnell, J.D. Bowman, P. Brindza, D.B. Brown, R.D. Carlini, G.D. Cates, B. Cavness, G. Clark, J.C. Cornejo, S. Covrig Dusa, M.M. Dalton, C.A. Davis, D.C. Dean, W. Deconinck, J. Diefenbach, K. Dow, J.F. Dowd, J.A. Dunne, D. Dutta, W.S. Duvall, J.R. Echols, M. Elaasar, W.R. Falk, K.D. Finelli, J.M. Finn, D. Gaskell, M.T.W. Gericke, J. Grames, V.M. Gray, K. Grimm, F. Guo, J. Hansknecht, D.J. Harrison, E. Henderson, J.R. Hoskins, E. Ihloff, K. Johnston, D. Jones, M. Jones, R. Jones, M. Kargiantoulakis, J. Kelsey, N. Khan, P.M. King, E. Korkmaz, S. Kowalski, A. Kubera, J. Leacock, J.P. Leckey, A.R. Lee, J.H. Lee, L. Lee, Y. Liangv, S. MacEwan, D. Mack, J.A. Magee, R. Mahurin, J. Mammei, J.W. Martin, A. McCreary, M.H. McDonald, M.J. McHugh, P. Medeiros, D. Meekins, J. Mei, R. Michaels, A. Micherdzinska, A. Mkrtchyan, H. Mkrtchyan, N. Morgan, J. Musson, K.E. Mesick, A. Narayan, L.Z. Ndukum, V. Nelyubin, Nuruzzaman, W.T.H. van Oers, A.K. Opper, S.A. Page, J. Pan, K.D. Paschke, S.K. Phillips, M.L. Pitt, M. Poelker, J.F. Rajotte, W.D. Ramsay, W.R. Roberts, J. Roche, P.W. Rose, B. Sawatzky, T. Seva, M.H. Shabestari, R. Silwal, N. Simicevic, G.R. Smith, S. Sobczynski, P. Solvignon, D.T. Spayde, B. Stokes, D.W. Storey, A. Subedi, R. Subedi, R. Suleiman, V. Tadevosyan, W.A. Tobias, V. Tvaskis, E. Urban, B. Waidyawansa, P. Wang, S.P. Wells, S.A. Wood, S. Yang, S. Zhamkochyan, R.B. Zielinski.(2015). The  $Q_{weak}$  experimental apparatus. *Nuclear Instruments and Methods in Physics Research*, 781, 105-133.
- [f] C.E. Carlson.(2013). Nucleon Measurement at the Precision frontier. *AIP Conference Proceedings*, 1563, 20-28
- [g] K.D. Paschke.(2013). Precision Electroweak Studies Using Parity Violation in Electron Scattering. *AIP Conference Proceedings*, 1563, 98-101
- [h] Henning Weier, Harald Krauss, Markus Rau, Martin Fürst, Sebastian Nauerth, Harald Weinfurter.(2011). Quantum eavesdropping without interception: an attack exploiting the dead time of single-photon detectors. *New Journal of Physics*, 13, 1-11
- [i] K. Aulenbacher.(2013). The MESA Accelerator. *AIP Conference Proceedings*, 1563, 5-12

- [j] D. Becker, K. Gerz, S. Baunack, K.S. Kumar, F.E. Maas.(2013). P2 - The Weak Charge of the Proton. *AIP Conference Proceedings*, 1563, 78-86
- [k] D. Nährmann.(1994). *Das Große Werkbuch Elektronik*; Franzis'
- [l] C. Schenk.(1980). *Halbleiter-Schaltungstechnik*. 6. Auflage; Springer
- [m] J. Diefenbach.(2010). *Bestimmung der Analysierstärke des A4-Compton-Rückstreupolarimeters zur Messung der longitudinalen Spinpolarisation des MAMI-Elektronenstrahls*. Unv. Diss.; Uni Mainz
- [n] K. Grimm.(2002). *Aufbau eines Lichtauslesesystems für ein Bleifluorid-Kalorimeter zur Messung der Paritätsverletzung in der elastischen Elektronenstreuung*. Unv. Diss.; Uni Mainz
- [o] Wiki des F-Praktikums der Uni Mainz;  
<https://www-fp.physik.uni-mainz.de/index.php>

## D. Danksagung

An dieser Stelle möchte ich mich bei allen Personen, die mich bei der Erstellung dieser Arbeit unterstützt haben danken. Durch die ausgesprochen nette und freundliche Atmosphäre innerhalb der Arbeitsgruppe ist das vergangene Jahr sehr rasch vorbeigegangen.

Als erstes bedanke ich mich bei Professor Dr. Frank Maas, der mir die Möglichkeit gegeben hat diese Masterarbeit anzufertigen. Er war stets am Fortschritt meiner Arbeit interessiert und hat in einigen Gruppenterminen einen Großteil seiner Zeit auf meine kleinen Problemchen verwendet. Besonders bedanken möchte ich mich bei Dr. Sebastian Baunack, meinem direkten Betreuer, der stets Zeit für mich hatte und mir in allen Phasen der Arbeit durch seine Hilfe und Interesse über einige Herausforderungen half.

Ich bedanke mich bei allen anderen aus dem Büro und der Arbeitsgruppe, also Boris, Dominik, Jürgen, Kathrin und Yoshio mit denen man stets über alle Fragen und Probleme, die sich bei der Erstellung der Arbeit ergaben, reden konnte.

Weiterer besonderer Dank gilt im Institut Prof. Dr. Werner Lauth und meinem Vater, für ihre Unterstützung bei allen Problemen technischer und elektronischer Natur.

Meiner Mutter danke ich für die generelle Unterstützung bei meiner Entscheidung zum Physikstudium.

Als letztes danke ich noch allen Freunden und Kommilitonen die mich auf meinem Studienweg begleitet haben und immer für Abwechslung sorgten, wenn diese einmal nötig war.