

DIREKTE MESSUNG DER  
DRIFTGESCHWINDIGKEIT IM GAS DER  
CMS-MYONENKAMMERN MIT VDC-KAMMERN

von  
**Carsten Heidemann**

Diplomarbeit in Physik

vorgelegt der

Fakultät für Mathematik, Informatik und Naturwissenschaften  
der RWTH Aachen

im Juni 2010

angefertigt im

III. Physikalischen Institut A

Prof. Dr. Thomas Hebbeker

Zweitgutachter

Dr. Oliver Pooth

© 2010



---

## Zusammenfassung

Diese Arbeit behandelt die Softwareentwicklungen, Firmware, Datennahme- und Steuerungsprogramme für das VDC<sup>1</sup>-System des III. Physikalischen Instituts A der RWTH Aachen. Das VDC-System ist eine Kombination von verschiedenen Messgeräten zur Charakterisierung des Füllgases der Myonenkammern des CMS am LHC. Durch die genaue Messung der Veränderung der Driftgeschwindigkeit von Elektronen innerhalb des Gases wird eine genaue Ortsauflösung in den Myonenkammern des CMS erreicht. Diese wiederum verbessert die Impulsauflösung der detektierten Myonen und erlaubt somit eine präzisere Analyse der Kollisionsereignisse.

Die nötigen Programme für den dauerhaften Betrieb des VDC-Systems, die von mir erstellt wurden, werden beschrieben und erklärt. Neben der Entwicklung der Firmware für die neuen Koinzidenz- und Diskriminatormodule wurden auch Datennahme- und Steuerungsprogramme entwickelt.

Ziel dieser Arbeit war es, den großen Schritt von Messungen mit dem Prototypen einer VDC hin zur ersten simultanen Messung mit sechs VDCs zu machen.

## Abstract

This thesis describes the software development for a VDC system designed at RWTH Aachen University. The development includes firmware, data acquisition and control software. The VDC system is a combination of different measurement instruments in order to characterize the gas of the CMS muon chambers at LHC. Monitoring the variation of the electrons drift speed in the gas leads to a exact spatial resolution. This also improves the momentum resolution of the detected muons and it allows a more precise analysis of the collision events.

All programs which are required for a continuous operation, written by myself, are described and explained here. Besides the development of the firmware for new hardware modules, software for data acquisition and control was written.

The aim of this work was to master the step from simple measurements with a VDC prototype to simultaneous measurements with six VDCs.

---

<sup>1</sup>VDC steht für Velocity Drift Chamber (Geschwindigkeitsdriftkammer)



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>VI</b>
<b>Verzeichnis der Listings</b>	<b>VII</b>
<b>1. Einleitung</b>	<b>1</b>
<b>2. Physikalische Grundlagen</b>	<b>8</b>
2.1. Standardmodell der Teilchenphysik . . . . .	8
2.1.1. Fermionen - Bausteine unserer Materie . . . . .	8
2.1.2. Fundamentale Kräfte . . . . .	9
2.1.3. Higgs Mechanismus . . . . .	11
2.2. Experimentell zu beantwortende Fragen . . . . .	11
2.3. Physik der Gasetektoren . . . . .	12
2.3.1. Ionisation . . . . .	12
2.3.2. Bewegung der Elektronen . . . . .	13
2.3.3. Das elektrische Feld im Gasetektor . . . . .	14
<b>3. VDC-System Aachen</b>	<b>15</b>
3.1. VDC - Prinzip . . . . .	15
3.2. Komponenten des VDC-Systems . . . . .	17
3.2.1. VDC . . . . .	17
3.2.1.1. Kammer . . . . .	17
3.2.1.2. Triggereinheit . . . . .	18
3.2.1.3. Anodenvorverstärker . . . . .	19
3.2.1.4. Hochspannungsfiler und Spannungsteiler . . . . .	20
3.2.2. Versorgungseinheiten . . . . .	22
3.2.2.1. Hochspannungen . . . . .	22
3.2.2.2. Niederspannungen . . . . .	22
3.2.2.3. Weitere Einheiten . . . . .	23
3.2.3. Triggerlogik . . . . .	23
3.2.4. Überwachungs- und Steuerungsgeräte . . . . .	26
3.2.4.1. Gasmatrix . . . . .	26
3.2.4.2. Druckdosen . . . . .	27

---

3.2.4.3.	Druck- und Flussregler . . . . .	27
3.2.4.4.	Temperatursensoren . . . . .	28
3.3.	Überwachung . . . . .	29
<b>4.</b>	<b>Datennahme- und Steuersoftware</b>	<b>30</b>
4.1.	Hoch- und Niederspannungen . . . . .	30
4.2.	VME-Zugriffe . . . . .	31
4.2.1.	TDC-Auslese . . . . .	31
4.2.2.	Zählerauslese . . . . .	32
4.3.	OneWire-Buszugriffe . . . . .	33
<b>5.</b>	<b>Universaleinschübe</b>	<b>34</b>
5.1.	Anforderungen . . . . .	34
5.2.	Hardware . . . . .	35
5.3.	Bedienung . . . . .	35
5.4.	Firmware . . . . .	36
5.4.1.	Programmstruktur . . . . .	36
5.4.2.	Ausgegliederte und mehrfach genutzte Submodule . . . . .	37
5.4.2.1.	Zähler . . . . .	37
5.4.2.2.	RS232 . . . . .	39
5.4.2.3.	VME / VME IRQ . . . . .	39
5.4.2.4.	Entpreller . . . . .	39
5.4.3.	Performance des Designs . . . . .	39
<b>6.</b>	<b>Testmöglichkeiten</b>	<b>44</b>
6.1.	Koinzidenztests . . . . .	44
6.2.	Diskriminatortests . . . . .	45
6.3.	Timingtest . . . . .	45
6.4.	Zählertests . . . . .	46
6.5.	RS232-Test . . . . .	46
6.6.	Ferndiagnose . . . . .	46
<b>7.</b>	<b>Ergebnisse der Messungen</b>	<b>47</b>
7.1.	Hochspannungsmessungen . . . . .	47
7.2.	Ratenmessungen . . . . .	48
7.3.	Temperaturmessungen für die Ventilüberwachung . . . . .	50
7.4.	Driftgeschwindigkeitsmessungen . . . . .	51
7.5.	Abschließende Bemerkungen . . . . .	55
<b>8.</b>	<b>Fazit und kritische Bewertung</b>	<b>61</b>
	<b>Literaturverzeichnis</b>	<b>63</b>

---

<b>Eidesstattliche Erklärung</b>	<b>65</b>
<b>A. VHDL-Quelltexte</b>	<b>i</b>

## Abbildungsverzeichnis

1.1. Übersicht LHC Gelände . . . . .	1
1.2. Compact Muon Solenoid . . . . .	2
1.3. Schnittbild durch das Myonsystem . . . . .	4
1.4. Mehrlagige Myonenkammer . . . . .	5
1.5. Querschnitt einer Driftzelle mit Feld . . . . .	6
3.1. Kammeraufbau . . . . .	16
3.2. VDC-Schrank . . . . .	18
3.3. Triggerbox . . . . .	19
3.4. Anodenverstärker . . . . .	20
3.5. HV-Kathoden-Filterbox, Spannungsteiler und HV-Anoden-Filter . . . . .	20
3.6. Schaltbild der Kathodenhochspannung . . . . .	21
3.7. Schaltbild der Anodenhochspannung . . . . .	21
3.8. Hochspannungsrahmen . . . . .	22
3.9. Versorgungsrahmen . . . . .	24
3.10. Logikplan des Triggers . . . . .	25
3.11. Gasmatrix . . . . .	26
3.12. Druckdose . . . . .	27
3.13. Druck- und Flussregler an der Kammer . . . . .	28
4.1. Flussdiagramm der HV-Auslese . . . . .	30
4.2. Flussdiagramm der TDC-Auslese . . . . .	32
4.3. Flussdiagramm der Zählerauslese . . . . .	32
4.4. Flussdiagramm der Temperatúrauslese . . . . .	33
5.1. Frontblenden der Universalmodule . . . . .	41
5.2. Platinen des Prototyps der Koinzidenz COINC3x2 . . . . .	42
5.3. Übersichtsbild des Hauptmoduls . . . . .	43
6.1. Schema der Testsignalerzeugung . . . . .	44
7.1. Graph der Hochspannung ohne Auffälligkeiten . . . . .	48
7.2. Graph des Stroms ohne Auffälligkeiten . . . . .	49
7.3. Graph der Hochspannung mit Störung . . . . .	50
7.4. Graph des Stroms mit Störung . . . . .	51



---

7.5. Graph der Over/Understates mit Störung . . . . .	52
7.6. Graph der Raten an den SiPMs, SiPM-V Scan . . . . .	53
7.7. Graph der Raten an den SiPMs, Stabilitätsmessung . . . . .	54
7.8. Graph der Raten an den SiPMs , SiPM2-V Scan . . . . .	55
7.9. Graph der Raten an den SiPMs, SiPM1-V Scan . . . . .	56
7.10. Graph der Raten an den SiPMs, Messung mit optischen Testpulsen . . . . .	57
7.11. Graphen der Raten an der Anode, Anoden-V Scan . . . . .	57
7.12. Graph zur Bestimmung der Heizzeit . . . . .	58
7.13. Graph zur Bestimmung der Totzeit . . . . .	58
7.14. Driftzeitspektrum von VDC4 . . . . .	59
7.15. Driftzeitspektrum von VDC6 . . . . .	59
7.16. Simultane Messung von Driftgeschwindigkeiten . . . . .	60

## Tabellenverzeichnis

2.1. Tabelle der Leptonen . . . . .	8
2.2. Tabelle der Quarks . . . . .	9
2.3. Tabelle der fundamentalen Kräfte . . . . .	10

---

## Verzeichnis der Listings

5.1. Funktionaler VHDL-Codeauszug des Zählersubmoduls . . . . .	37
A.1. Vollständiger VHDL-Quelltext des Zählersubmoduls . . . . .	i
A.2. Vollständiger VHDL-Quelltext des Entprellersubmoduls . . . . .	iii
A.3. Vollständiger VHDL-Quelltext des RS232-Interface Submoduls . . . . .	iv
A.4. Vollständiger VHDL-Quelltext des RS232-Receive-Subsubmoduls . . . . .	ix
A.5. Vollständiger VHDL-Quelltext des RS232-Send-Subsubmoduls . . . . .	xi
A.6. Vollständiger VHDL-Quelltext des VME-Submoduls . . . . .	xiii



# 1. Einleitung

Die Teilchenphysik beschäftigt sich mit den teils noch unbekanntenen Bausteinen der bekannten Materie. Dazu werden entweder hochenergetische Teilchen zur Kollision gebracht oder aber einfach der Himmel beobachtet, da aus dem Weltraum ebenfalls hochenergetische Teilchen zu uns kommen. Das CERN ist die Europäische Organisation für Elementarteilchenphysik. Ein Forschungsschwerpunkt dieser Organisation liegt auf dem LHC, dem in puncto Schwerpunktsenergie (14 TeV als Endziel) derzeit leistungsfähigsten verfügbaren Ringbeschleuniger für Proton-Proton-Kollisionen.

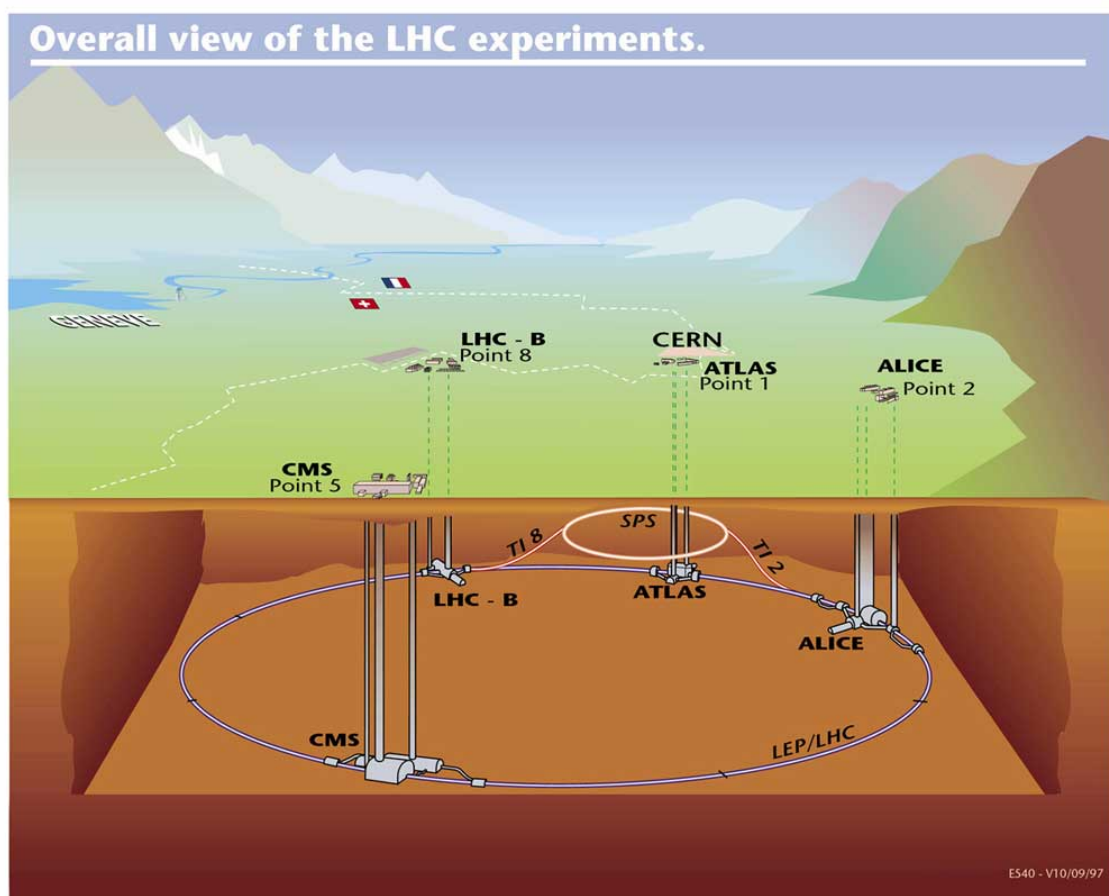


Abbildung 1.1.: Ansicht des LHC mit den Experimenten und einem Vorbeschleuniger [1]

Es handelt sich hier um den größten Ringbeschleuniger der Welt. Aufgrund seines Umfangs

von 26,659 km werden sich mit den derzeitigen Magneten Schwerpunktsenergien bei Proton-Proton-Kollisionen von bis zu 14 TeV erreichen lassen. An den Strahlkreuzungspunkten finden die Teilchenkollisionen statt und daher sind dort die verschiedenen Teilchendetektoren aufgestellt. An vier der Strahlkreuzungen sind komplexe Detektoren zum Nachweis von Reaktionsprodukten aufgestellt. Die vier großen Experimente sind ALICE (A Large Ion Collider Experiment), ATLAS (A Toroidal Lhc ApparatuS), CMS (Compact Muon Solenoid) und LHCb( Large Hadron Collider beauty experiment), teilweise durch kleinere Experimente ergänzt. Bei ALICE handelt es sich um einen Vielweckdetektor, der auf die Untersuchung von Schwerionenkollisionen optimiert ist. Das LHCb Experiment wird insbesondere die CP-Verletzung bei B-Mesonen vermessen. Bei ATLAS und CMS handelt es sich um Universaldetektoren mit zwiebelschalenförmigem Aufbau. Diese beiden Experimente sind so gebaut, dass sie jeweils die Ergebnisse des anderen überprüfen können.

Der Compact Muon Solenoid Detektor (Abb. 1.2) besteht aus einem Solenoid-Magneten und

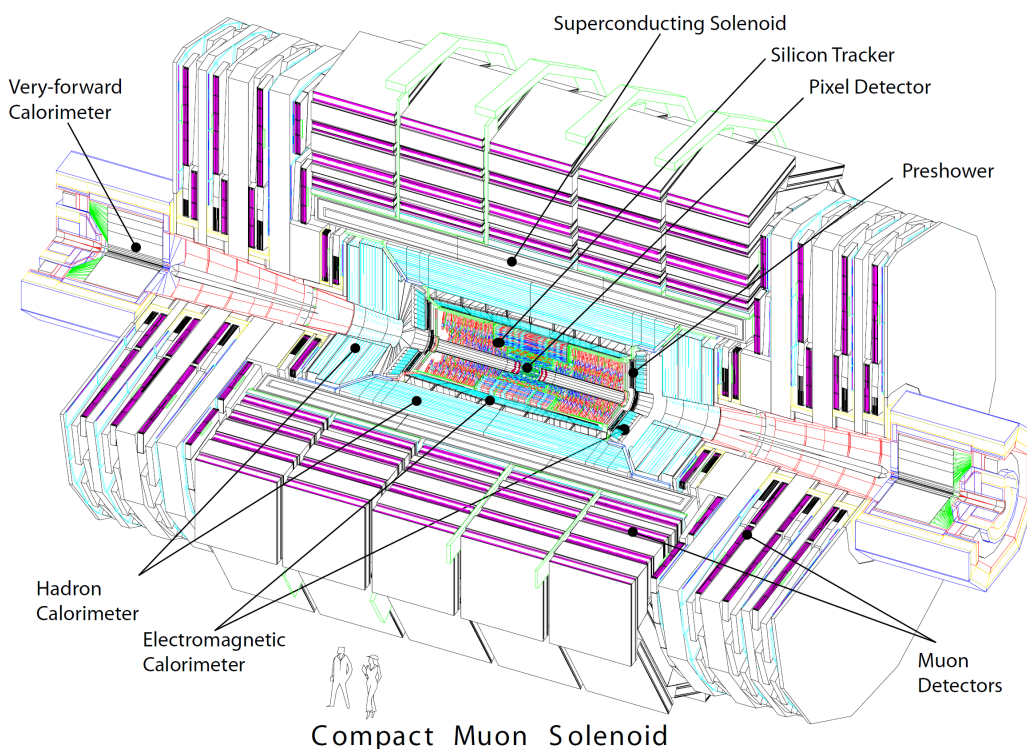


Abbildung 1.2.: Schematische Darstellung<sup>[2]</sup> des CMS Detektors mit farblicher Hervorhebung einzelner Systeme. Die Strahlkollisionen finden im Mittelpunkt statt. Mit einem Durchmesser von 15 m und einer Masse von 12500 t wird der Detektor dennoch als kompakt bezeichnet. Die schweren Einzelteile können entlang der Strahlachse verschoben werden, um den Zugang zu den Detektorkomponenten zu ermöglichen.

verschiedenen Subdetektorsystemen. Der Magnet sorgt mit seinem im Inneren der Spule 4 T starken Magnetfeld für eine Krümmung der Bahnen aller geladenen Teilchen, wodurch eine Impulsmessung ermöglicht wird. Der dem Kollisionspunkt am nächsten liegende Subdetektor ist ein Siliziumpixeldetektor, welcher geladene Teilchen mit einer sehr hohen Ortsauflösung nachweisen kann. Die Ortsauflösung liegt in der Größenordnung von 0,01 mm. Dahinter folgt ein größerer Siliziumstreifendetektor mit einer etwas größeren Ortsauflösung von 0,02 – 0,1 mm. Die beiden Siliziumdetektoren dienen zur Vertex- und Impulsrekonstruktion der geladenen Teilchen. Eine Schicht weiter aussen folgt das elektromagnetische Kalorimeter mit Blei-Wolframat-Kristallen zur Messung der Energie von Photonen, Elektronen und Positronen und geladenen Hadronen. Die Auflösung des elektromagnetischen Kalorimeters ist von der Energie des zu vermessenden Teilchens abhängig<sup>2</sup>, ein 120 GeV Elektron wird mit einem Fehler von 0,45% vermessen. Bei geladenen Hadronen wird aber nur ein Teil der Energie im elektromagnetischen Kalorimeter deponiert. Dahinter befindet sich das hadronische Kalorimeter, welches zur Energiebestimmung von geladenen und neutralen hadronischen Teilchen bestimmt ist. Auch hier ist die Auflösung von der Energie abhängig<sup>3</sup>. Es besteht aus mehreren Lagen Messingplatten und Szintillatoren. Die Messingplatten dienen als Absorbermaterial und erzeugen Teilchenschauer. Diese werden mit den Szintillatoren nachgewiesen. Hinter den Kalorimetern folgt der supraleitende Spulenkörper des Magneten. Das äußerste Subdetektorsystem ist das Myonsystem. Dessen Aufgabe ist die Identifikation und die Messung von Impuls und Ladung der Myonen. Dazu liegen im Rückführjoch des Magneten zahlreiche Driftkammern und RPCs<sup>4</sup>. In den beiden Endkappen wurden CSCs<sup>5</sup> verbaut. Myonen und Neutrinos sind i.d.R. die einzigen Teilchen, die bis hier vordringen können. Da aber Neutrinos nur vernachlässigbar selten wechselwirken, dienen die Driftkammern in der Praxis ausschließlich zur Ortsbestimmung des Myonendurchgangs. Aufgrund des Magnetfeldes bewegen sich die Myonen auf gekrümmten Bahnen. Aus dem Krümmungsradius lässt sich dann der Impuls der einzelnen Myonen bestimmen.

Die Identifikation wird dadurch erreicht, dass nur Myonen bis zu den Myonen-Kammern durchkommen, da fast alle anderen Teilchen, die ein Signal erzeugen könnten, auf Grund des langen Weges durch massives Detektormaterial absorbiert wurden. Die angestrebte relative Impulsauflösung liegt bei 10%. Zusätzlich müssen alle Myonen erfasst werden können, daher darf es keine toten Bereiche geben, durch die ein Myon unbeobachtet entkommen könnte (Abb. 1.3). Da besonders auch die Ereignisse mit mehreren Myonen interessant sind, sollte die Einzeleffizienz  $\sigma$  hoch sein, da sich die Gesamteffizienz  $\sigma_{Ges}$  eines Multi-Myon-Ereignisses als Produkt der Einzeleffizienzen berechnet. Für den „goldenen“ Zerfallskanal eines Higgsteilchens ( $H_0 \rightarrow Z_0 Z_0 \rightarrow \mu^+ \mu^- \mu^+ \mu^-$ ) läge bei 90% Einzeleffizienz die Rekonstruktionswahrscheinlichkeit bei nur noch 66%. Da dieser Prozess nur einen sehr geringen Wirkungsquerschnitt  $\sim 100$  fb bei einer Schwerpunktsenergie von 14 TeV besitzt, ist eine hohe Effizienz bei der Detektion erforderlich. Um die gewünschte Impuls-

---

<sup>2</sup> $(\sigma/E)^2 = \left(2.8\%/\sqrt{E}\right)^2 + (0,12/E)^2 + (0,30\%)^2$ [2]

<sup>3</sup> $(\sigma/E)^2 = \left(100\%/\sqrt{E}\right)^2 + (4,5\%)^2$ [3]

<sup>4</sup>Resistive plate chamber, eine andere Art von Gasdetektor

<sup>5</sup>Cathode strip chamber, ebenfalls ein Gasdetektor

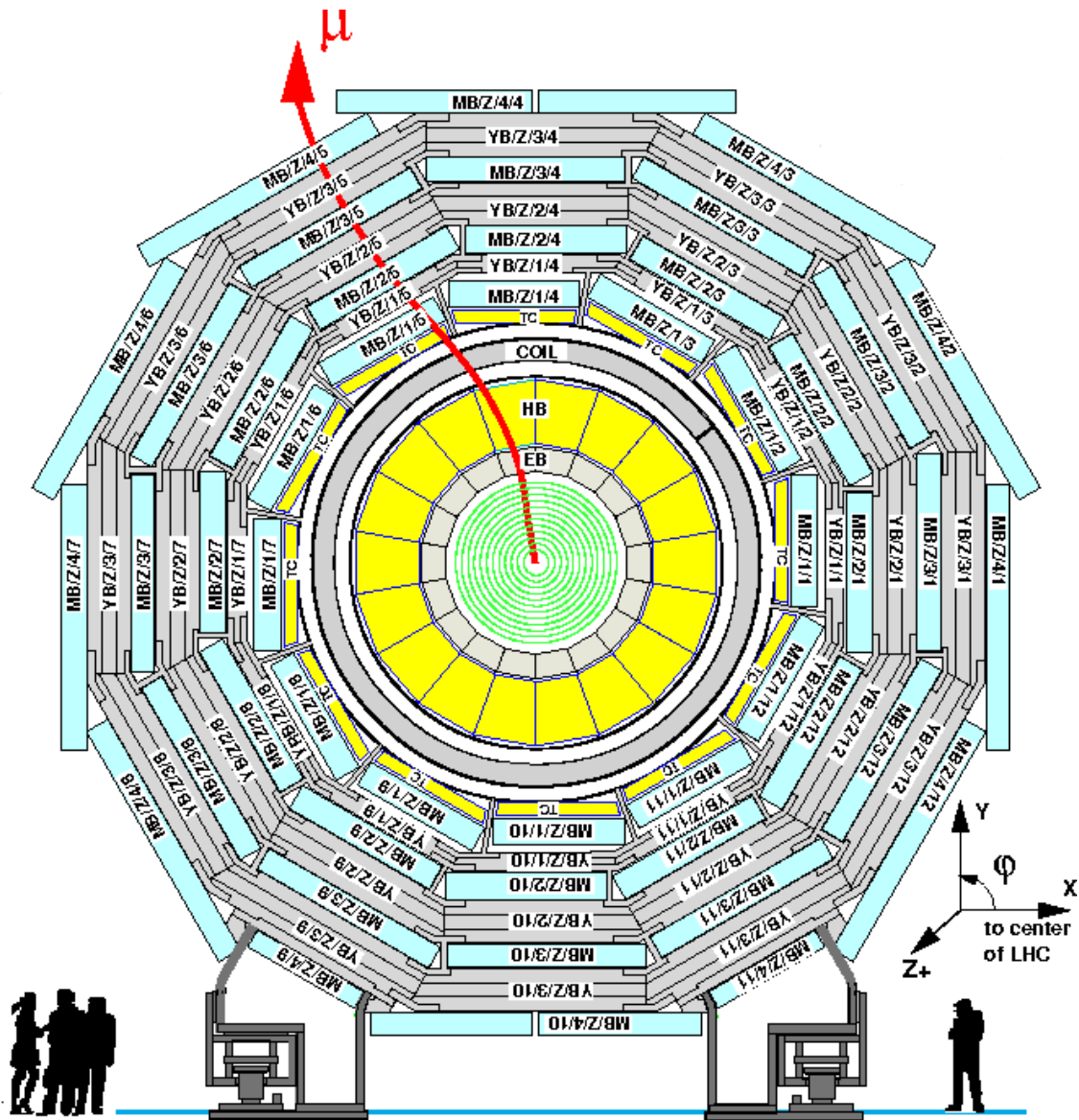


Abbildung 1.3.: Querschnitt[2] durch das Myonsystem in der zentralen Region mit deutlich sichtbarem Überlapp der Kammern, um alle Myonen erfassen zu können. Die mit YB bezeichneten Stahlelemente bilden das Rückführjoch für die solenoide Spule.

auflösung zu erreichen, muss eine Ortsauflösung des Myonendurchgangs von  $\sim 100 \mu\text{m}$  erreicht werden, da die zu ermittelnden Sagittae der Flugbahnen der Myonen recht klein sind. Für ein 1 TeV Myon beträgt sie für 1 m Spurlänge etwa  $150 \mu\text{m}$ <sup>6</sup>. Zur Vermessung der Myonen werden verschiedene Arten von Gasedektoren eingesetzt. Diese basieren auf der Ionisation des Füllgases durch ein durchfliegendes Teilchen. Die entstehenden Ionen und Elektronen driften wegen des vorhandenen elektrischen Feldes in entgegengesetzte Richtungen. Die Elektronen driften schnell in Richtung Anode und die Ionen aufgrund ihrer hohen Masse wesentlich langsamer in Richtung Kathode. Durch die Verwendung eines dünnen Anodendrahtes lassen sich aus wenigen Primär-

<sup>6</sup>Näherung für  $s = \frac{0,3B[T]L[m]^2}{8p_T[\text{GeV}]}$ ,  $L \gg s$  mit  $L$  Spurlänge,  $B$  Magnetfeld,  $p_T$  Myonimpuls



elektronen mit Hilfe der Gasverstärkung viele Sekundärelektronen erzeugen, die ein messbares Signal beim Auftreffen auf der Anode liefern. Die im zentralen Bereich des CMS-Detektors (CMS-

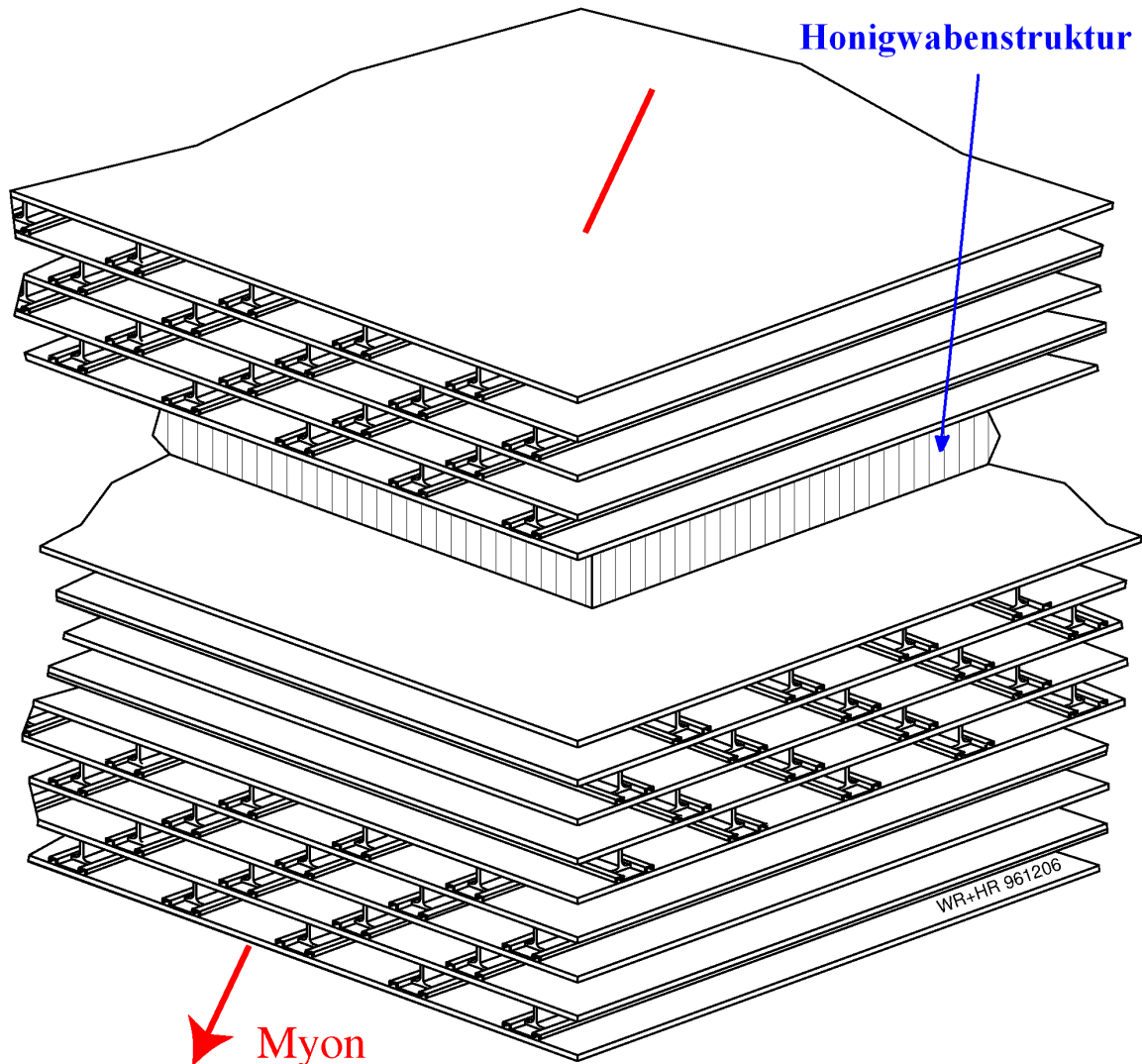


Abbildung 1.4.: Darstellung[4] einer Myonenkammer mit drei Superlayern, jeweils bestehend aus vier Lagen Driftzellen in verschiedener Ausrichtung. Die 12 cm dicke Honigwabenstruktur in der Mitte vergrößert den Abstand zwischen erstem und drittem Superlayer und dient zur Verbesserung der Winkelauflösung. Gleichzeitig wird damit die Kammer versteift. Entlang der Peripherie der Bienenwabe sind optische Elemente zur Vermessung der Kammerposition, Kabel, Rohre, Drucksensoren, Befestigungselemente und die Ausleseelektronik untergebracht.

Barrel) verbauten Myonenkammern bestehen aus mehreren Lagen von Driftzellen. Diese bieten einen relativ großen Bereich mit konstanter Driftgeschwindigkeit und eine kleine Zone um den Anodendraht herum mit Gasverstärkung für die Erzeugung eines messbaren Signals. Aus der Triggerzeit  $t_0$ , die dem Zeitpunkt der Entstehung der primären Ionisation gleichzusetzen ist und der Ankunftszeit  $t_1$  der Elektronen am Anodendraht kann mittels des Integrals  $\int_{t_0}^{t_1} v_d(t)dt$  der Abstand der primären Ionisation vom Anodendraht berechnet werden.

Eine Myonenkammer besteht aus drei Superlayern. Diese sind um  $90^\circ$  gegeneinander ver-

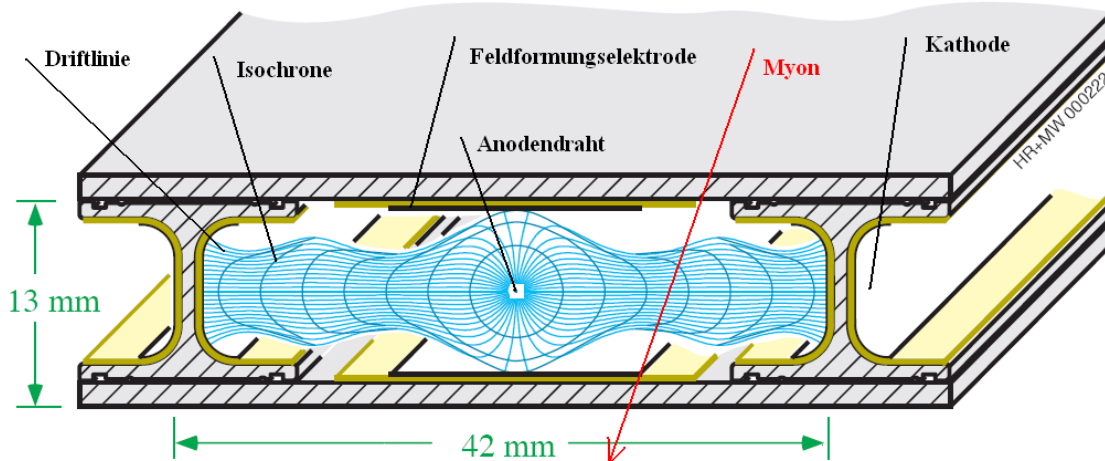


Abbildung 1.5.: Querschnitt[4] einer Driftzelle mit Driftlinien und Isochronen für alle Elektronenbahnen, die zur Anode führen. Hier dargestellt mit einem Magnetfeld  $B = 0$  in der Zelle. Die Kammer wird mit einem Argon-Kohlendioxidgemisch (85 / 15) bei minimalem Überdruck betrieben. Die Betriebsspannungen betragen +3600 V für die Drähte, +1800 V für die Feldformungselektroden und -1200 V für die Kathoden. Die daraus resultierenden Feldstärken innerhalb der Zelle von 200 – 250 V/mm sorgen für eine fast konstante Driftgeschwindigkeit von  $55 \mu\text{m/ns}$ .

dreht (Abb. 1.4). Ein Superlayer besteht aus vier Lagen leicht versetzt angeordneter Driftzellen (Abb. 1.5). Durch diesen Aufbau lassen sich x- und y-Position eines Myonendurchgangs innerhalb der Kammer bestimmen. Die Auflösung, die eine einzelne Zelle erreicht, ist besser als  $250 \mu\text{m}$ . Da in einer Kammer mehrere Zellen von einem Myon getroffen werden, verbessert sich die Auflösung auf die benötigten  $100 \mu\text{m}$ . Dabei fließen verschiedenartige Störungen mit in die Messung ein. Bei großen Driftstrecken ist die Diffusion der driftenden Elektronen dominant. Zusätzlich spielt die Fluktuation in der Anzahl der primär erzeugten Elektronen eine Rolle. Die mechanische Stabilität, u.a. das Durchhängen der Drähte unter Einfluss der Schwerkraft, sowie die Fertigungspräzision haben Einfluss auf die Driftstrecke und das elektrische Feld. Die Auflösengenauigkeit der Ausleselektronik  $\sigma_T \sim 1 \text{ ns}$  spielt ebenfalls eine große Rolle. Diese Einflüsse sind aber zeitlich weitestgehend konstant. Diese zeitliche Konstanz ist jedoch bei den Versorgungsspannungen und den Driftgeschwindigkeiten nicht gegeben, deshalb müssen diese überwacht werden. Die Überwachung der Hochspannungen lässt sich direkt an den Netzteilen realisieren, da diese Geräte Daten über Strom und Spannung zur Verfügung stellen. Bei der Driftgeschwindigkeit gestaltet sich die Überwachung nicht ganz so leicht. Speziell zu diesem Zweck wurden „Velocity-Drift-Chambers“ (VDC) gebaut. Es gibt für jedes der fünf Hauptelemente („Räder“) des CMS Detektors eine VDC und eine zusätzliche VDC als Reserve. Die Priorität bei der Messung liegt auf der Erfassung von Änderungen der Driftgeschwindigkeit  $v_D$ . Die VDCs erlauben auch eine absolute Messung der Driftgeschwindigkeit. Mit den sechs VDCs ist auch eine Untersuchung der systematischen Fehler

---

der Driftgeschwindigkeitsmessung durch die VDCs möglich. Das elektrische Feld in der VDC ist einstellbar und ermöglicht die Bestimmung der Abhängigkeit  $v_D(E)$ . Jedes der Räder besitzt eine eigene Druck- und Flussregelung. In einem Rad sind jeweils 50 Myonenkammern montiert. Bei Verwendung nur einer VDC würde man etwa zehn Tage brauchen, um das Gas jeder Myonenkammer einmal zu vermessen. Durch die Verwendung von fünf VDCs kann die Zeit für einen Durchlauf auf etwa 50 Stunden reduziert werden. Für eine regelmäßige Überprüfung reicht dieser Zyklus aus. Bei Auffälligkeiten einer Myonenkammer kann deren Gas aber auch zeitnah untersucht werden.

Im nächsten Kapitel werden kurz die Grundlagen der Teilchenphysik und der Gasetektorphysik dargestellt. Danach folgt die detaillierte Beschreibung des VDC-Systems.

## 2. Physikalische Grundlagen

### 2.1. Standardmodell der Teilchenphysik

#### 2.1.1. Fermionen - Bausteine unserer Materie

Zahlreiche Experimente zur Untersuchung von Strukturen kleiner als  $10^{-18}$  m haben gezeigt, dass Materie aus 12 verschiedenen elementaren Teilchen besteht. Diese werden als Fermionen bezeichnet. Sie lassen sich in zwei Gruppen einteilen, die Leptonen (Tab. 2.1) und die Quarks (Tab. 2.2). Weiterhin gibt es zu jedem Teilchen ein Antiteilchen mit demselben Spin, aber invertierten ladungsähnlichen Quantenzahlen. Die Leptonen und Quarks lassen sich in drei Generationen<sup>7</sup> einteilen. Die uns umgebende Materie besteht jedoch nur aus Teilchen der ersten Familie, nämlich Elektronen sowie zwei Bindungszuständen von Up- und Downquarks, dem Proton bestehend aus Up-Up-Down und dem Neutron aus Down-Down-Up. Elektronen wie auch die anderen Leptonen können isoliert auftauchen. Innerhalb eines Atoms formen Protonen und Neutronen den Kern, während die Elektronen die Schale bilden. Elektronen besitzen die kleinste Ladung, die ein isoliertes Teilchen haben kann. Daher bezeichnet man diese Ladung als Elementarladung  $e$ , sie entspricht  $-1.6022 \cdot 10^{-19}$  C. Die Ladung der Quarks beträgt  $2/3 e$  für das up-Quark und  $-1/3 e$  für das down-Quark. So ergibt sich für das positiv geladene Proton eine Ladung von  $1 e$ . Das neutrale Neutron hat die Ladung  $0 e$ .

Zusätzlich zu der elektrischen Ladung besitzen die Quarks eine Farbladung (rot, blau und grün).

Name	Symbol (Anti-)	Familie (Flavor)	elektrische Ladung	Farbladung	Masse [MeV]
Elektron	$e^- (e^+)$	I	$-1 e$	-	0,511
Elektronneutrino	$\nu_e (\bar{\nu}_e)$	I	$0 e$	-	$< 2 \cdot 10^{-6}$
Myon	$\mu^- (\mu^+)$	II	$-1 e$	-	105,7
Myonneutrino	$\nu_\mu (\bar{\nu}_\mu)$	II	$0 e$	-	$< 0,19$
Tau	$\tau^- (\tau^+)$	III	$-1 e$	-	1776,8
Tauneutrino	$\nu_\tau (\bar{\nu}_\tau)$	III	$0 e$	-	$< 18.2$

Tabelle 2.1.: Tabelle der Leptonen [5]

Um ein stabiles Hadron zu bilden, muss die Farbe weiß sein. Damit ergeben sich mindestens zwei Möglichkeiten, um ein stabiles Hadron zu bilden. Zum einen aus drei Quarks oder drei Antiquarks, da (anti)rot, (anti)blau und (anti)grün zusammen weiß ergeben, zum anderen aus Quark

<sup>7</sup>alternativ auch als Familie oder Flavour bezeichnet

Name	Symbol (Anti-)	Familie (Flavor)	elektrische Ladung	Farb- ladung	Masse [MeV]
up	$u$ ( $\bar{u}$ )	I	$+2/3 e$	rgb	$1,5 - 3,3$
down	$d$ ( $\bar{d}$ )	I	$-1/3 e$	rgb	$3,5 - 6,0$
strange	$s$ ( $\bar{s}$ )	II	$-1/3 e$	rgb	$104^{+26}_{-34} - 130$
charm	$c$ ( $\bar{c}$ )	II	$+2/3 e$	rgb	$1270^{+70}_{-110}$
bottom	$b$ ( $\bar{b}$ )	III	$-1/3 e$	rgb	$4200^{+170}_{-70}$
top	$t$ ( $\bar{t}$ )	III	$+2/3 e$	rgb	$\sim 171200 \pm 2100$

Tabelle 2.2.: Tabelle der Quarks [5]

und Antiquark, da z. B. rot und antirot auch weiß ergeben. Hadronen aus drei Quarks werden als Baryonen bezeichnet und solche aus zwei Quarks als Mesonen. Theoretisch ist es auch möglich aus vier oder fünf Quarks ein weißes Objekt zu bilden, aber alle experimentellen Untersuchungen dazu haben bisher keine eindeutigen Anhaltspunkte für solche Hadronen geliefert.

Die Leptonen (Quarks) der zweiten und dritten Generation sind denen der ersten sehr ähnlich. Sie unterscheiden sich lediglich in ihrer Masse, Lebensdauer und Leptonen-(Baryonen-)Quantenzahl. Ein Myon ist ein schweres Elektron mit begrenzter Lebensdauer und anderer Leptonenfamilienzahl, besitzt aber sonst die gleichen Eigenschaften.

Ein Art von Teilchen, die bislang noch keine Erwähnung fand, sind die Neutrinos. Zu jedem geladenen Lepton (Elektron, Myon und Tau) gibt es ein zugehöriges Neutrino. Sie sind elektrisch neutral und besitzen einen sehr kleinen Wirkungsquerschnitt und eine sehr geringe Masse. Daher lassen sie sich nur sehr schwer nachweisen. Selbst speziell für den Neutrinonachweis gebaute Experimente weisen trotz des gewaltigen Flusses von Neutrinos nur wenige Neutrinos pro Tag nach.

Jedes Teilchen ist mit bestimmten Quantenzahlen verknüpft. Beispiele für additive Quantenzahlen sind Leptonenzahl, Leptonenfamilienzahl und Flavorquantenzahl. Bei einer Wechselwirkung bleibt die Summe jeder additiven Quantenzahl erhalten. Jedes Teilchen des Standardmodells besitzt ein Antiteilchen mit gleicher Masse und gleichem Spin, aber entgegengesetzten Quantenzahlen. Zum Beispiel hat das Myon  $\mu^-$  die Ladung  $-e$  und die Leptonquantenzahl  $+1$ . Das zugehörige Antimyon  $\mu^+$  besitzt die Ladung  $+e$  und hat eine Leptonquantenzahl von  $-1$ .

Der Spin, ebenfalls eine Quantenzahl, ist eine weitere Teilcheneigenschaft und wird in Einheiten von  $\hbar$  ausgedrückt. Fermionen haben halbzahligen und Bosonen ganzzahligen Spin.

### 2.1.2. Fundamentale Kräfte

In der klassischen Physik werden Kräfte und Wechselwirkungen durch Felder, z.B. das elektromagnetische Feld, übertragen. Diese Methode versagt allerdings auf Teilchenebene, da sie zahlreiche

Phänomene nicht beschreiben kann. Daher wurden neue Theorien entwickelt, nämlich die QFD<sup>8</sup> und die QCD<sup>9</sup>. Diese Theorien werden im Standardmodell der Teilchenphysik zusammengefasst und liefern Erklärungen für drei der vier bekannten Kräfte (vgl. 2.3), nämlich die schwache, die elektromagnetische und die starke Wechselwirkung. Die Gravitation wird durch die Quantenfeldtheorie nicht erklärt. In der Feldtheorie werden die Wechselwirkungen über Eichbosonen<sup>10</sup> übertragen. Ob ein Teilchen einer Wechselwirkung unterliegt, hängt nur davon ab, ob es mit dem entsprechenden Eichboson wechselwirkt oder nicht.

Kraft	Eichboson	Masse [GeV]	Reichweite
starke	Gluonen	0	$10^{-15}$
elektromagnetische	Photon	0	$\infty$
schwache	$W^\pm, Z^0$	80, 91	$10^{-18}$
gravitative	Graviton	0	$\infty$

Tabelle 2.3.: Tabelle der fundamentalen Kräfte: Das Graviton wurde bislang nicht nachgewiesen, es ist jedoch mit einer Masse von 0 postuliert. Die Masse der Gluonen ist von der Theorie mit 0 vorhergesagt. Alle Experimente zu Gluonen konnten bislang keine von 0 verschiedene Masse nachweisen.

Die elektromagnetische Wechselwirkung wird durch das Photon übertragen. Das Photon besitzt keine Ruhemasse und bewegt sich daher mit Lichtgeschwindigkeit. Die Reichweite der elektromagnetischen Wechselwirkung ist unendlich. Die Kraft nimmt zwar mit zunehmender Distanz ab, verschwindet aber nie ganz. Das Photon wechselwirkt mit allen geladenen Teilchen, ist aber selbst elektrisch neutral und kann daher weder die Ladung eines Teilchens ändern noch mit sich selbst wechselwirken.

Die schwache Wechselwirkung wird durch die  $W^\pm$  und  $Z^0$  Bosonen vermittelt. Sie besitzen eine „schwache“ Ladung, die es ihnen erlaubt, untereinander zu wechselwirken. Die elektrisch geladenen W-Bosonen können auch mit Photonen wechselwirken. Da sie große Ruhemassen besitzen, ist die Reichweite der schwachen Wechselwirkung nur sehr klein. Im Standardmodell werden die elektromagnetische und die schwache Wechselwirkung zur elektroschwachen Wechselwirkung vereint.

Die starke Wechselwirkung betrifft nur Quarks und Gluonen. Die Reichweite ist extrem kurz, sie reicht gerade einmal soweit, um die drei Quarks innerhalb eines Protons zusammenzuhalten. Mathematisch lassen sich diese drei fundamentalen Wechselwirkungen als Symmetriegruppen darstellen. Das direkte Produkt zweier Symmetriegruppen  $SU(2) \times U(1)$  beschreibt die elektroschwache Wechselwirkung mit vier elektroschwachen Austauscheteilchen ( $\gamma, Z, W^+, W^-$ ). Die starke Wechselwirkung lässt sich durch eine spezielle unitäre Gruppe mit drei Dimensionen ( $SU(3)$ ) beschreiben und enthält damit acht Austauscheteilchen, die Gluonen.

<sup>8</sup>Quantenflavordynamik, beschreibt die elektromagnetische und die schwache, zusammen kurz die elektroschwache Wechselwirkung

<sup>9</sup>Quantenchromodynamik, beschreibt die starke Wechselwirkung

<sup>10</sup>Auch als Austauscheteilchen oder Feldquanten bezeichnet

Die gravitative Wechselwirkung ist nicht durch das Standardmodell beschrieben und das zugehörige Boson, das Graviton, wurde noch nicht entdeckt. Die Reichweite ist unendlich, aber die Stärke ist extrem schwach im Vergleich zu den anderen drei Wechselwirkungen. Dennoch reicht sie aus, um Schwarze Löcher entstehen und wachsen zu lassen.

### 2.1.3. Higgs Mechanismus

Die Massen der Teilchen variieren über mindestens 11 Größenordnungen. Das leichteste massenbehaftete Teilchen ist das Elektronenneutrino mit einer Masse  $m$  von weniger als 2 eV[5], das schwerste bekannte Teilchen ist das Top-Quark mit  $m = 171,2$  GeV[5]. Das Standardmodell beschreibt weder diese Varianz der Massen noch deren Ursprung. Eine mögliche und vieldiskutierte Lösung ist der Vorschlag von P. Higgs. Er erklärt die Massen der Teilchen mit Hilfe eines zusätzlichen Feldes und dem zugehörigen Eichboson, dem Higgs-Boson. Dieses Feld wechselwirkt mit allen anderen Teilchen und die Wechselwirkung ist proportional zur Masse der Fermionen.

## 2.2. Experimentell zu beantwortende Fragen

Trotz der vielen Experimente, die das Standardmodell als zutreffend überprüft haben, bleiben bislang noch einige Fragen offen[6], einige davon sind:

- Warum gibt es drei Generationen von Teilchen?
- Lassen sich die fundamentalen Kräfte in einer Theorie vereinigen?
- Gibt es Alternativen zum Finetuning<sup>11</sup>?
- Woraus besteht die dunkle Materie?

Um einen Teil der vielen offenen Fragen der Teilchenphysik beantworten zu können, wurde der LHC gebaut. Die dort erreichbaren Energien reichen auf jeden Fall aus, um die Higgs-Theorie zu überprüfen. Aus verschiedenen Beobachtungen konnten bereits obere und untere Massengrenzen für das Higgsteilchen bestimmt werden. Die derzeit besten Daten vom Tevatron<sup>12</sup> liefern eine obere Grenze von 186 GeV[7]. Eine untere Grenze wurde z. B. experimentell am LEP<sup>13</sup> festgestellt. Massen unterhalb von 114,4 GeV[8] konnten ausgeschlossen werden. Die am LHC erreichbaren Schwerpunktsenergien von 14 TeV reichen aus, um ein Standardmodell-Higgsteilchen zu bestätigen oder aber auszuschließen.

---

<sup>11</sup>Beschreibt das Phänomen, dass die freien Parameter des Standardmodells sehr exakt angepasst werden müssen, damit sie die Beobachtungen erklären.

<sup>12</sup>Ringbeschleuniger für  $p\bar{p}$ -Kollisionen am Fermilab mit maximaler Schwerpunktsenergie von 1,96 TeV.

<sup>13</sup>Large Electron Positron Collider, Vorgänger des LHC

## 2.3. Physik der Gasdetektoren

Gasdetektoren lassen sich zum Nachweis von ionisierender Strahlung verwenden. Innerhalb der Detektoren werden durch Stoßionisation Elektronen und Ionen erzeugt. Durch ein angelegtes elektrisches Feld lassen sich Elektronen und Ionen trennen und bis zu einer durch Stöße mit anderen Gasmolekülen begrenzten Geschwindigkeit beschleunigen. Mittels der Gasverstärkung können diese an der Anode ein messbares elektrisches Signal erzeugen.

### 2.3.1. Ionisation

Es gibt verschiedene Arten von Ionisation. Die Autoionisation von hoch angeregten Atomen und die Feldionisation durch ein sehr starkes elektrisches Feld spielen für den Betrieb von Gasdetektoren keine Rolle. Lediglich bei der Wahl der elektrischen Feldstärke sollte die Feldionisation, um diese an den Elektroden und im Gas zu verhindern, berücksichtigt und tunlichst vermieden werden, da eine Feldionisation im Detektor die Messung unmöglich machen würde. Die für Gasdetektoren entscheidende Ionisationsart ist die Stoßionisation durch ionisierende Strahlung. Dabei stoßen die ionisierenden Teilchen mit den Gasmolekülen (Atomen) zusammen. Ist die dabei auf das Molekül (Atom) übertragene Energie groß genug, wird ein Elektron aus der Aussenschale freigesetzt. Ionisierende Teilchen mit hoher Energie erzeugen in der Regel eine Vielzahl von Elektronen-Ionen-Paaren. Der Energieverlust pro Weglänge für schwere Teilchen, z.B. den Myonen in den Driftkammern, lässt sich mittels der Bethe-Bloch-Formel[9] berechnen:

$$-\frac{dE}{dx} = 4\pi N_A r_e^2 m_e c^2 z^2 \frac{Z}{A} \rho \frac{1}{\beta^2} \left( \ln \frac{2m_e c^2 \gamma^2 \beta^2}{I} - \beta^2 - \frac{\delta}{2} \right) \quad (2.1)$$

mit  $x =$  Weglänge,  $N_A =$  Avogadrokonstante,  $r_e =$  Klassischer Elektronradius,  $m_e =$  Elektronenmasse,  $z =$  Ladung des ionisierenden Teilchens in Einheiten der Elementarladung,  $Z, A =$  Ordnungszahl und Atomgewicht des Absorbers,  $\rho =$  Dichte des Absorbers,  $\gamma =$  Lorentz-Faktor  $= E/m_0 c^2$ ,  $\beta = v/c$ ,  $I =$  mittlere Ionisationsenergie  $\approx 16Z^{0.9}$  eV für  $Z > 1$  und  $\delta =$  Fermi-Plateau  $\approx 2\ln\gamma + c$ ,  $c$  Materialkonstante. Eine grobe Näherung ist durch

$$-\frac{dE}{dx} \approx 2 \cdot \frac{\text{MeV} \cdot \text{cm}^2}{\text{g}} \cdot \rho \quad (2.2)$$

gegeben. Für Elektronen muss die Bethe-Bloch-Formel angepasst werden, da sich der Energieübertrag aufgrund der gleichen Masse der Stoßpartner ändert[9]:

$$-\frac{dE}{dx} = 4\pi N_A r_e^2 m_e c^2 \frac{Z}{A} \frac{1}{\beta^2} \left( \ln \frac{\gamma m_e c^2}{2I} - \beta^2 - \frac{\delta^*}{2} \right) \quad (2.3)$$



mit denselben Größen wie zuvor,  $\delta^*$  hat einen etwas anderen Wert als  $\delta$ . Zusätzlich beschreibt diese Formel nur bis zu einer kritischen Energie

$$E \ll E_C = \frac{710 \text{ MeV}}{Z + 0,92} \quad (2.4)$$

den Energieverlust ausreichend genau. Für höhere Energien spielt die Bremsstrahlung eine bedeutende Rolle, für  $E > E_C$  dominiert die Bremsstrahlung. Für Argon ( $Z = 18$ ) erhält man eine kritische Energie  $E_C$  von 37,5 MeV. Da die ionisierenden Elektronen der VDC aus Sr90-Quellen stammen, haben sie eine maximale Energie von 2,28 MeV und daher dominiert der Energieverlust durch Ionisation. Die Ionisationsenergie zur Erzeugung eines Elektron-Ion-Paares beträgt für Gase allgemein  $\sim 30$  eV und für Argon etwa 26 eV. Um permanente Entladungen im Edelgas, die durch UV-Anregung und Photoeffekte auf der Kathode entstehen, zu verhindern, verwendet man ein mehratomiges Löschgas, wie z.B.  $\text{CH}_4$ ,  $\text{C}_4\text{H}_{10}$  oder  $\text{CO}_2$ . Diese Gase absorbieren die Anregungsenergien und verhindern somit die dauerhafte Entladung. Das benötigte Mischungsverhältnis variiert je nach Anwendung. Die Myonenkammern am CMS verwenden eine Argon-Kohlendioxid-Mischung mit einem  $\text{CO}_2$ -Anteil von 15%.

### 2.3.2. Bewegung der Elektronen

Die durch Ionisation erzeugten freien Elektronen stoßen mit den Gasmolekülen und verlieren dadurch überschüssige Energie, bis diese soweit reduziert ist, dass Energieaufnahme und -abgabe im Mittel gleich groß sind. Dieses Gleichgewicht wird durch die Maxwell-Boltzmann-Verteilung beschrieben[10].

$$F(\epsilon) = A \sqrt{\frac{\epsilon}{kT}} \cdot e^{-\epsilon/kT} \quad (2.5)$$

$$F(v) = \left(\frac{m_e}{2\pi kT}\right)^{3/2} \cdot e^{-m_e v^2/2kT} \quad (2.6)$$

Die mittlere thermische Energie beträgt  $\epsilon = (3/2)kT$ , die mittlere Geschwindigkeit ergibt sich dann zu

$$\bar{v} = \sqrt{\frac{3kT}{m_e}} \quad (2.7)$$

Da die Bewegungsrichtung aufgrund der mehrfachen Stöße zufällig ist, lässt sich die Diffusion der Elektronen durch

$$\frac{dN}{N} = \frac{1}{\sqrt{4\pi Dt}} \cdot e^{-x^2/4Dt} dx \quad (2.8)$$

beschreiben. Diese Gleichung gibt an, welcher Anteil der Elektronen sich nach einer Zeit  $t$  im Bereich  $dx$  im Abstand  $x$  vom Ursprungsort befindet. Dabei ist  $D$  die Diffusionskonstante und hängt direkt mit der freien Weglänge  $\lambda$  ( $D = (\bar{v} \cdot \lambda)/3$ ) zusammen. Legt man nun ein Feld an,

so bewegen sich die Elektronen aufgrund dieser Kraft mit einer anderen Geschwindigkeit. Diese kann durch

$$v_d = \frac{2}{3} \frac{eE}{m} \left\langle \frac{\lambda}{\bar{v}} \right\rangle + \frac{1}{3} \frac{eE}{m} \left\langle \frac{d\lambda}{d\bar{v}} \right\rangle \quad (2.9)$$

beschrieben werden [11]. Mit  $\lambda = 1/(N\sigma(\epsilon))$  ( $N$  Teilchendichte,  $\sigma(\epsilon)$  Streuquerschnitt) und  $\bar{v} = \sqrt{2\epsilon/m}$  erhält man

$$v_d = \frac{1}{3} \frac{eE}{mM} \left( \left\langle \sqrt{\frac{2m}{\epsilon}} \cdot \frac{1}{\sigma(\epsilon)} \right\rangle + \left\langle \sqrt{\frac{2m}{\epsilon}} \cdot \epsilon \cdot \frac{d}{d\epsilon} \frac{1}{\sigma(\epsilon)} \right\rangle \right) \quad (2.10)$$

Da über die Energieverteilungen gemittelt wird, wird für kleine Driftfelder der Ausdruck in der Klammer konstant.

$$v_d \propto \frac{E}{N} \quad (2.11)$$

Für ein ideales Gas ( $pV = NkT$ ) lässt sich dann, da  $V$  und  $k$  konstant sind, näherungsweise schreiben

$$v_d \propto \frac{ET}{p} \quad (2.12)$$

Die Konstanthaltung der Driftgeschwindigkeit wäre technisch mit großem Aufwand durch die Regelung der drei Größen, elektrisches Feld, Temperatur und Druck möglich. Temperatur und Druck sind innerhalb einer Driftzelle in der Regel hinreichend konstant, während das elektrische Feld ortsabhängig ist. Die Konstanz ist bei den Myonenkammern nicht erforderlich, denn es reicht für die Myonenkammern aus, die Driftgeschwindigkeit zu kennen. Die Driftgeschwindigkeit lässt sich mit einer VDC bestimmen.

### 2.3.3. Das elektrische Feld im Gasdetektor

Im Inneren eines Gasdetektors wird ein elektrisches Feld erzeugt, welches die Elektronen und Ionen in entgegengesetzte Richtungen zieht und so die sofortige Rekombination der Elektronen und Ionen verhindert. Durch das anliegende elektrische Feld driften die Elektronen und die Ionen in entgegengesetzte Richtungen. Die negativ geladenen Elektronen bewegen sich auf die Anode zu und die positiven Ionen auf die Kathode und erzeugen beim Auftreffen dort ein messbares Signal. Bei einem Gasdetektor, der als Proportionalzähler dient, wird das Signal aufintegriert, sodass die primäre Ionisation gemessen wird. Bei Gasdetektoren, die nur zum Nachweis dienen, erzeugt man ein größeres und damit leichter messbares Signal mit Hilfe der Gasverstärkung. Dazu wird die Anode so konstruiert, dass in ihrer Nähe ein sehr starkes elektrisches Feld herrscht. Die einfachste Möglichkeit, dies zu erreichen, ist die Verwendung eines sehr dünnen Drahtes. Aufgrund der großen Feldstärke in unmittelbarer Drahtnähe werden die Elektronen so stark beschleunigt, dass sie weitere Elektronen mittels Stoßionisationen freisetzen. Die dabei entstehenden Elektronenlawinen sorgen beim Auftreffen auf die Anode für einen Ladungsfluss, der einen Spannungsabfall von einigen Millivolt erzeugt. Dieser Spannungsabfall wird über einen Kondensator in einen Verstärker eingekoppelt.

### 3. VDC-System Aachen

Die Aufgabe dieses Systems ist die kontinuierliche und genaue Messung der Driftgeschwindigkeit von Elektronen im Gas mit einem Fehler von weniger als 0,1% [12]. Dieses System ist nicht das erste seiner Art. Die früheren Systeme von Böhm [13]<sup>14</sup> und Baksay [14] eignen sich aber nicht für die Umgebung, in der die VDC betrieben werden soll. Die Kammer von Böhm erreicht nicht die hier benötigte Hochspannungsfestigkeit. Die Kammer von Baksay setzt Laser als Ionisationsquellen ein und lässt sich daher nur schwer handhaben. Die absolute Messung der Driftgeschwindigkeit erfolgt bei allen über die Messung der Zeitdifferenz zweier verschieden langer Driftvorgänge. An der Kammer sind zwei Sr90-Quellen montiert, die für die Auslösung der Driftvorgänge sorgen. Der räumliche Auslösebereich ist sehr eng begrenzt, um eine hohe Ortsauflösung für die Differenzstrecke der beiden Driftstrecken zu erhalten. Der Drift erfolgt in einem sehr homogenen Feld mit einer Feldstärke, die auch in den Driftzellen des Myonsystem vorkommt. Die Kammer erzeugt letztlich drei Ausgangssignale, nämlich zwei Signale von der Auslöseeinheit für die Teilchendurchgänge und ein Anodensignal der mittels Gasverstärkung vervielfältigten Driftelektronen.

Da das VDC-System selbst ein Überwachungs- und Kontrollgerät ist, sollte es niemals in einen nicht definierten und unprotokollierten Zustand geraten. Daher werden alle Komponenten so genau wie möglich überwacht.

#### 3.1. VDC - Prinzip

Die VDC soll dazu dienen, bei vorgegebenem homogenen elektrischen Feld die Driftgeschwindigkeit im Prüfgas zu messen. Der Driftvorgang wird durch Elektronen aus einem Beta-Strahler ausgelöst, die das Gas entlang ihrer Bahn ionisieren und auch eine externe Triggereinheit ansprechen. Die Driftzeit ist die Zeitdifferenz zwischen Eintreffen des Anodensignals und des Triggersignals. Die Bestimmung der Verarbeitungszeit des Anoden- bzw. Trigger-Signals sowie der Übertragungszeit durch alle Kabel trägt zum systematischen Fehler bei. Da weiterhin in der Nähe des notwendigen Anodendrahts das elektrische Feld inhomogen ist, um eine Gasverstärkung zu erreichen, bedient man sich eines Umwegs (Abb. 3.1). Man verwendet zwei Quellen, mit verschiedenen Abständen zur Anode und verwendet als Messgröße die Differenz der beiden Driftzeiten. Damit gehen Verarbeitungszeit und Kabellängen nicht mehr in die Messung ein, auch die Inhomogenitäten in Anodennähe spielen keine Rolle mehr für die Messung. Es kommt nur darauf an, das gewünschte homogene elektrische Feld in der Region zwischen beiden Quellen

<sup>14</sup>siehe auch die Diplomarbeiten [15] und [16]

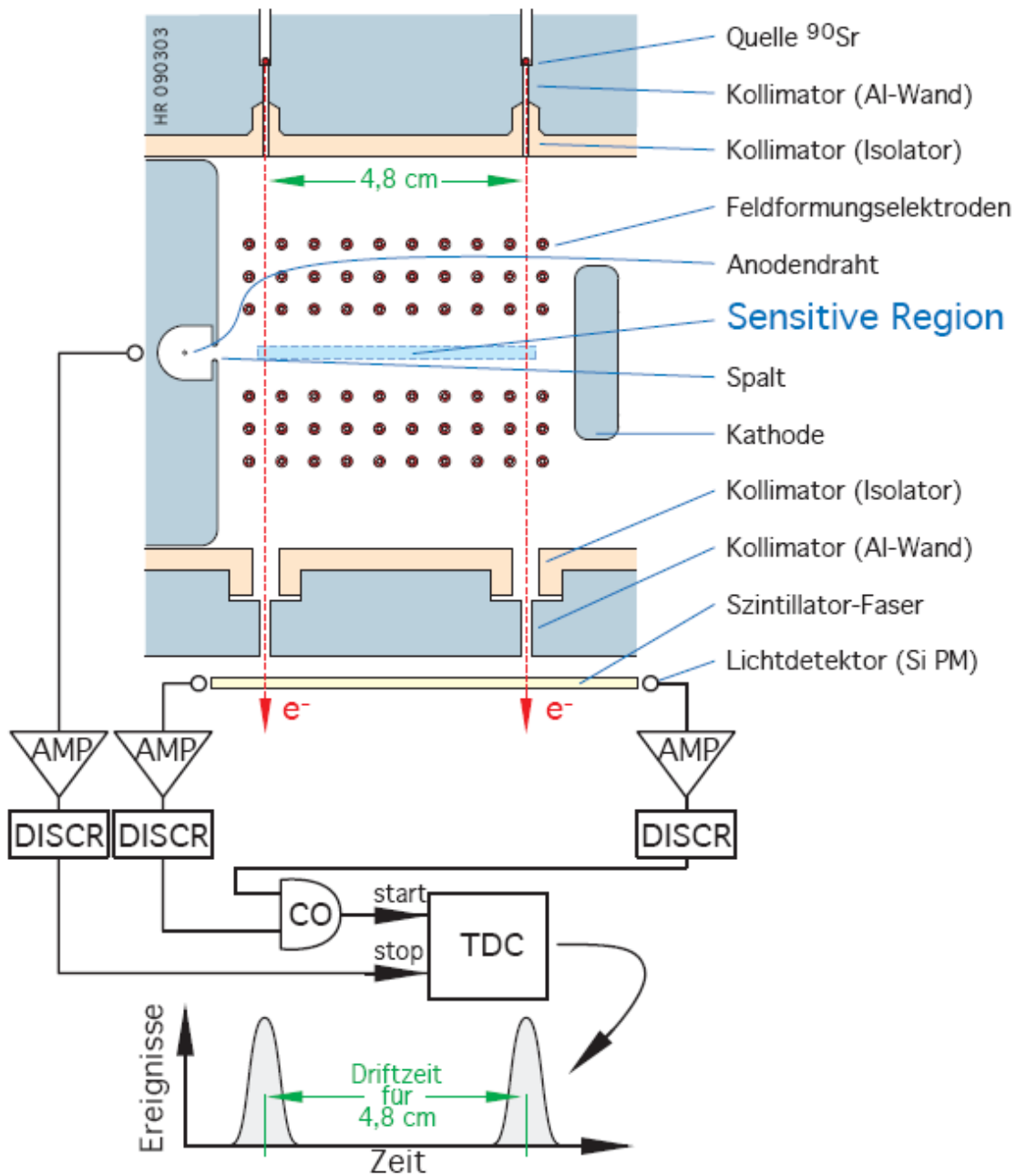


Abbildung 3.1.: Schematischer Aufbau[4] der Kammer, waagerechter Schnitt auf Höhe der Quellen und Auslassöffnungen

zu realisieren. Um die Differenzstrecke, also die Driftstrecke im homogenen Bereich des Feldes, präzise festzulegen, wird der Strahlendurchgang von Kollimatoren begrenzt. Der Startzeitpunkt des Driftvorgangs, gegeben durch den Teilchendurchgang durch die Kammer, wird durch die Lichtdetektoren in der Triggereinheit bestimmt. Der Endzeitpunkt wird durch das Signal der driftenden Elektronen an der Anode bestimmt. Die Differenz von End- und Startzeitpunkt wird als Ereignis in ein Histogramm eingetragen. Nach kurzer Messzeit kann man deutlich die zwei Spitzen erkennen. Mittels einer Gauss-Anpassung erhält man für jede Spitze einen Mittelwert.

Die Differenz aus den Mittelwerten liefert die Zeit, die die Elektronen brauchen, um die präzise bekannte Differenzstrecke zu driften. Die Driftgeschwindigkeit ist der Quotient von Differenzstrecke und Zeitdifferenz.

Um eine präzise Messung sicherzustellen, werden alle relevanten Parameter überwacht. Dafür wurden zahlreiche Komponenten sowohl in der mechanischen als auch in der elektrischen Werkstatt des III. Physikalischen Instituts A gebaut. Komponenten, die kommerziell in ausreichender Genauigkeit verfügbar waren, wurden gekauft.

Das System verfügt über zahlreiche Temperatursensoren, Gasdruckmesser, Gasdruck- und -flussregler, eine Möglichkeit für jede der sechs Kammern das Gas beliebig von Hand zu wählen und dies mit dem Computer zu erkennen, Hochspannungsversorgung mit integrierter Überwachung und eine Reihe von Versorgungseinheiten, die die Stromversorgung von und Kommunikation mit den angeschlossenen Teilen erlauben.

## 3.2. Komponenten des VDC-Systems

Das VDC-System (Abb. 3.2) besteht eigentlich aus 2 Schränken. Da diese aber sowohl logisch als auch mechanisch fest miteinander verbunden werden, wird immer nur von einem Schrank gesprochen. Die Hauptgruppen der in dem Schrank verbauten Hardware sind die VDCs, die Versorgungseinheiten, die Triggerlogik und die zusätzlichen Überwachungs- und Steuerungsgeräte.

### 3.2.1. VDC

#### 3.2.1.1. Kammer

Die Kammer (Abb. 3.1) besteht aus einem festen Aluminiumgehäuse. Im eingeschlossenen Gasvolumen von etwa einem Liter wird durch die Feldformungsgitter ein im sensitiven Bereich sehr homogenes einstellbares elektrisches Feld erzeugt. Bei der maximalen Kathodenspannung von  $-15$  kV beträgt die Feldstärke im sensitiven Bereich  $227$  V/mm. Der sensitive Bereich ist räumlich durch die Öffnung des Anodenraumes, die Kollimatoren der Quellenöffnung und der Siliziumphotomultiplier beschränkt und nimmt nur einen sehr kleinen Teil der Kammer ein. Das übrige Volumen der Kammer dient nur zur Feldformung. Nach rechts und links sorgen jeweils die drei Röhrchen des Gitters für ein ausreichend homogenes elektrisches Feld. Nach oben und unten wird die Homogenität des elektrischen Feldes über den im Vergleich zur Ausdehnung des sensitiven Bereichs sehr großen Abstand sichergestellt. Die erreichte Homogenität des elektrischen Feldes im sensitiven Bereich beträgt  $\delta E/E \approx 0,1\%$  [12].

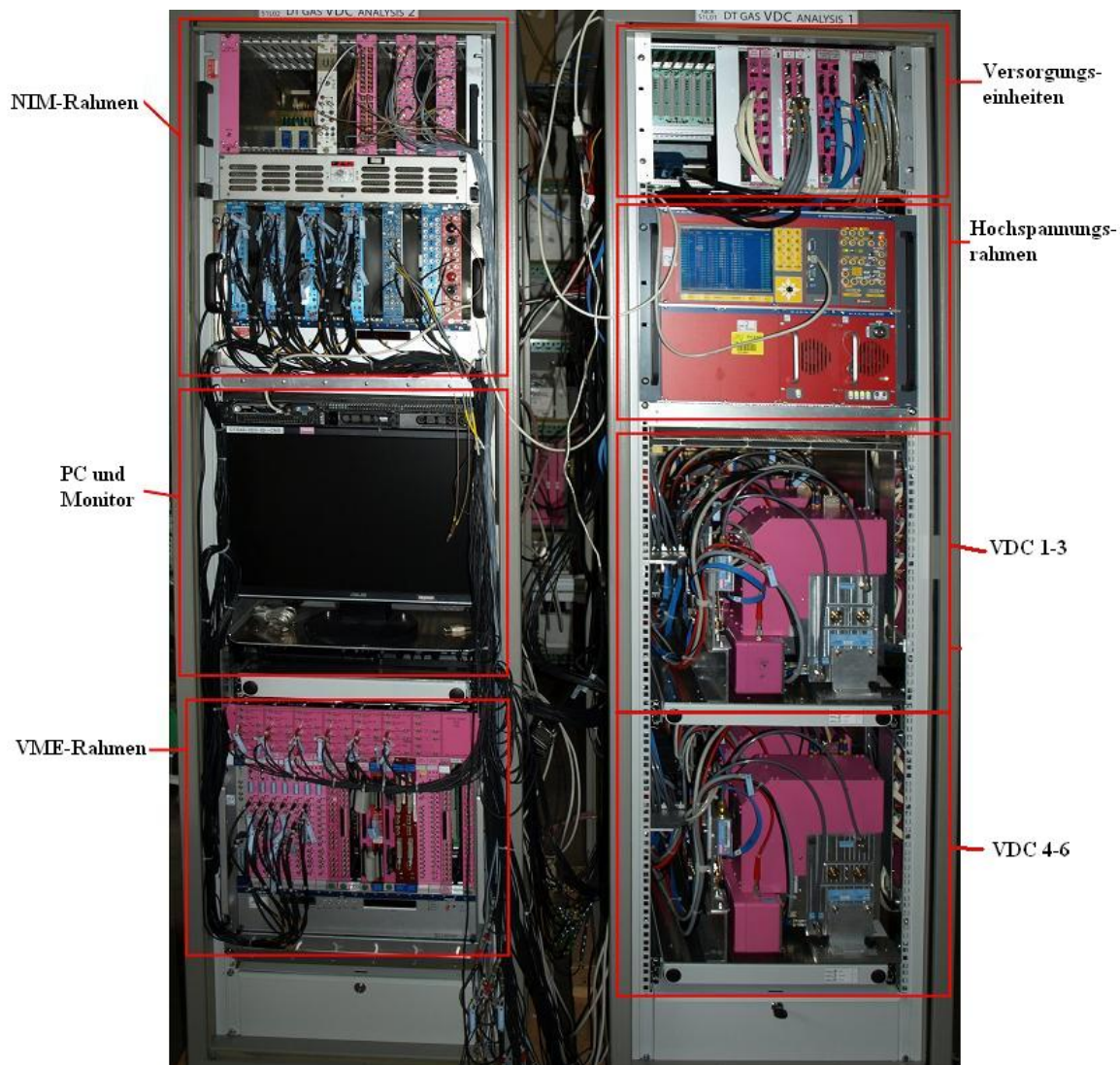


Abbildung 3.2.: VDC-Schrank mit markierten Hauptgruppen: NIM-Rahmen mit Teilen der Triggerlogik, PC mit Bildschirm zur Datenerfassung und Auswertung, VME-Rahmen mit dem Hauptteil der Triggerlogik, Versorgungseinheiten, Hochspannungsrahmen, die ersten drei VDCs auf der oberen Ebene und die zweiten drei auf der unteren Ebene. Im Schrank befinden sich zusätzlich noch u.a. Gasventile, Gassteuerelektronik und wassergekühlte Radiatoren der Luftkühlung. Die Türen sind im Foto nicht montiert.

### 3.2.1.2. Triggereinheit

Die Auslöseeinheit, die im Rahmen einer Diplomarbeit[17] entwickelt wurde, wird im folgenden als Triggerbox(Abb. 3.3) bezeichnet. Sie enthält eine Szintillatorfaser<sup>15</sup> mit einem quadratischen Querschnitt von  $1\text{ mm} \times 1\text{ mm}$ , an deren beiden Enden sich jeweils ein Siliziumphotomultiplier (SiPM)<sup>16</sup> befindet. Ein Teilchen, welches die Kammer verlässt und somit die hinter den Teilchen-

<sup>15</sup>Fasertyp: BCF-12 MC

<sup>16</sup>SiPM-Typ: Hamamatsu MPPC  $1\text{ mm}^2$  S10362-11-100C

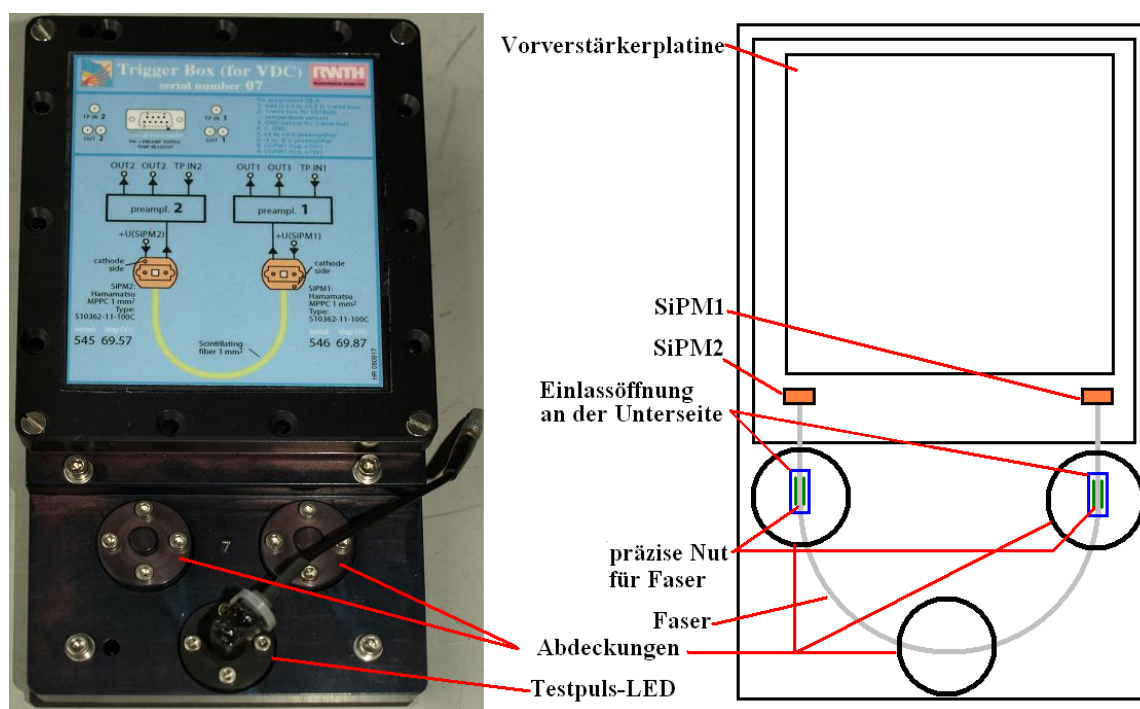


Abbildung 3.3.: Foto der Oberseite der Triggerbox. Die runden Abdeckungen sind zu Testzwecken gegen LEDs für optische Testpulse oder gegen radioaktive Quellen austauschbar. Auf dem Foto ist in der mittleren Position eine LED montiert. Die zwei anderen Positionen entsprechen den Austrittsschlitzen in der VDC. Die Einlassöffnungen befinden sich an der der VDC zugewandten Unterseite der Triggerbox und besitzen keine Trennung zwischen den Auslassöffnungen der VDC und der scintillierenden Faser in der Triggerbox.

auslassstellen befindliche Faser trifft, sollte also an den beiden SiPMs an den beiden Faserenden ein Signal erzeugen. Dieses noch schwache Signal wird durch die in die Triggerbox integrierten Verstärkerstufen soweit verstärkt, dass ein nachgeschalteter Diskriminator einen NIM-Puls erzeugen kann.

### 3.2.1.3. Anodenvorverstärker

Der Anodenvorverstärker (Abb. 3.4) befindet sich an der Unterseite der Kammer und erzeugt aus dem Signal am Anodendraht einen mittels Diskriminator umwandelbaren Puls. Dazu wird der Anodendraht mittels eines Kondensators auf der linken Seite der Platine in die Vorstufe des Verstärkers eingekoppelt. Die verwendeten Verstärkerbausteine<sup>17</sup> und die Vorstufe liefern in dieser Schaltung eine Verstärkung von Faktor 400. Zusätzlich lassen sich noch zwei Temperatursensoren auf der Platine unterbringen. Das Foto zeigt die Platine ohne diese optionalen Sensoren.

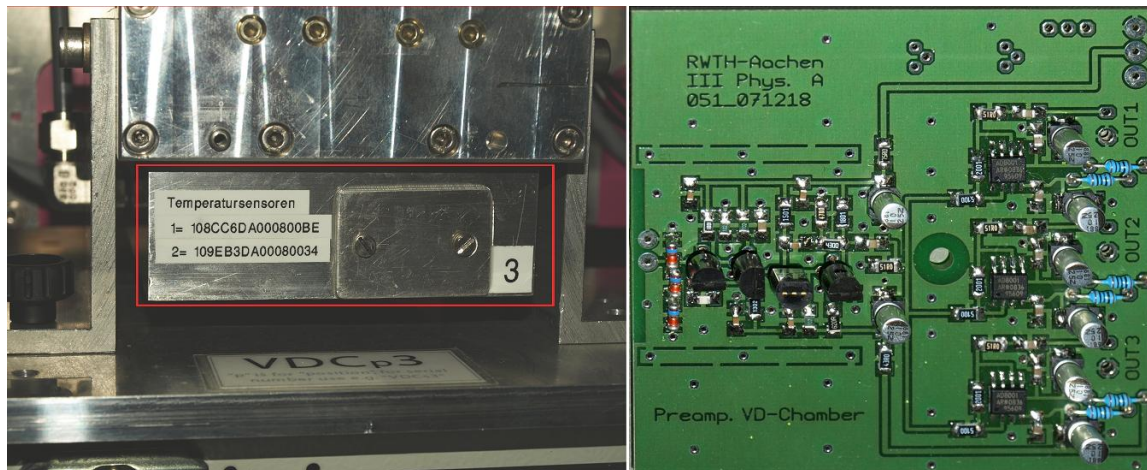


Abbildung 3.4.: Links ist an der Unterseite der VDC der Metalldeckel (rot umrandet im Bild) zu sehen, in dem der Anodenvorverstärker montiert ist. Auf der Platine sind rechts deutlich die drei parallelen Vorverstärkerbausteine für die drei Ausgänge zu erkennen und links auf der Platine befindet sich der Anschluss für die Anode.



Abbildung 3.5.: Links eine geöffnete und unvergossene Filterbox für die Kathodenhochspannung. Die an den Kammern montierten Metallboxen sind mit Metalldeckel versehen und zur Verbesserung der Spannungsfestigkeit mit einem Silikon gel (Wacker Sil gel) vergossen. In der Mitte ist der Spannungsteiler mit einem Widerstand von  $50 \text{ M}\Omega$  pro Stufe zur Versorgung des Gitters abgebildet. Bei  $-14 \text{ kV}$  fließt ein Strom von  $24 \mu\text{A}$  durch den Spannungsteiler. Der Strom der driftenden Ionen innerhalb der Kammer ist sehr viel geringer. Rechts ist der HV-Anoden-Filter zur Spannungsversorgung der Anode gezeigt. Die links in der Seite eingelassene Buchse dient zur Einspeisung eines Testpulses.

#### 3.2.1.4. HochspannungsfILTER und Spannungsteiler

Die Filterbox (Abb. 3.5, links) stabilisiert die eingespeiste Kathodenhochspannung durch Filterung der Spannungsspitzen. Die zwei Ausgänge (siehe auch Abb. 3.6) versorgen die Kathodenplatte im Inneren der Kammer und den Spannungsteiler. Der Spannungsteiler (Abb. 3.5, Mitte) versorgt die Feldformungsgitter mit passend geteilter Spannung, sodass im sensitiven Bereich ein sehr homogenes Feld entsteht. Die Anodenhochspannung wird über einen separaten und deutlich kleineren Filter (Abb. 3.5, rechts) eingespeist. Dieser enthält zusätzlich noch einen Testpulsengang (siehe auch Abb. 3.7), mit dem sich besonders die Signalkette vom Einspeisepunkt

<sup>17</sup>Typ: AD8001 AR



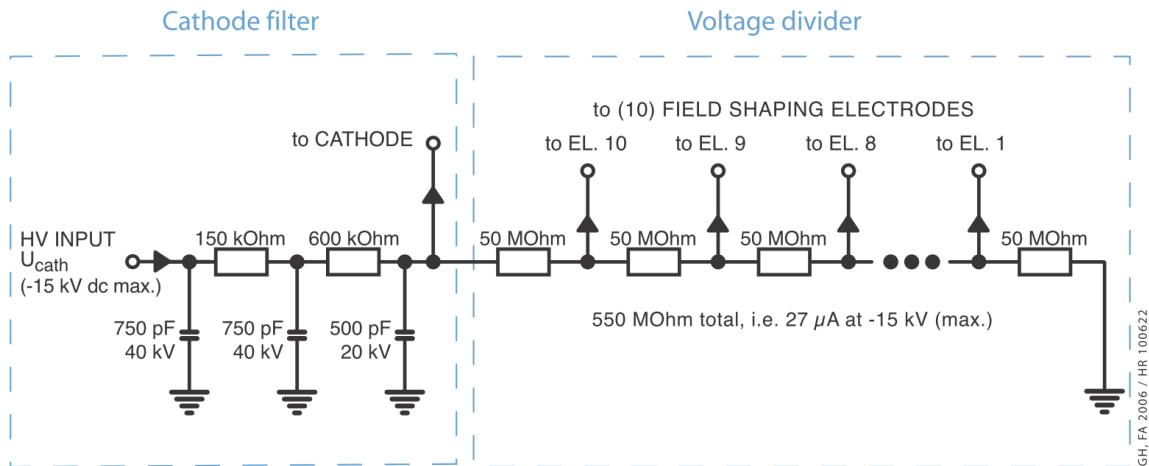


Abbildung 3.6.: Schaltbild[4] der Kathodenspannung. Links ist der Filter zum Glätten der Hochspannung abgebildet. Sowohl die Versorgung der Kathodenplatte als auch die des Spannungsteilers werden am selben Punkt abgegriffen. Der Spannungsteiler hat einen Gesamtwiderstand von 550 MΩ und versorgt die einzelnen Feldformungsgitter. Der Stromverbrauch des Spannungsteilers ist sehr viel größer als der Stromverbrauch im Inneren der Kammer, so dass sich kleine Schwankungen im Inneren kaum im Kathodenstrom erkennen lassen.

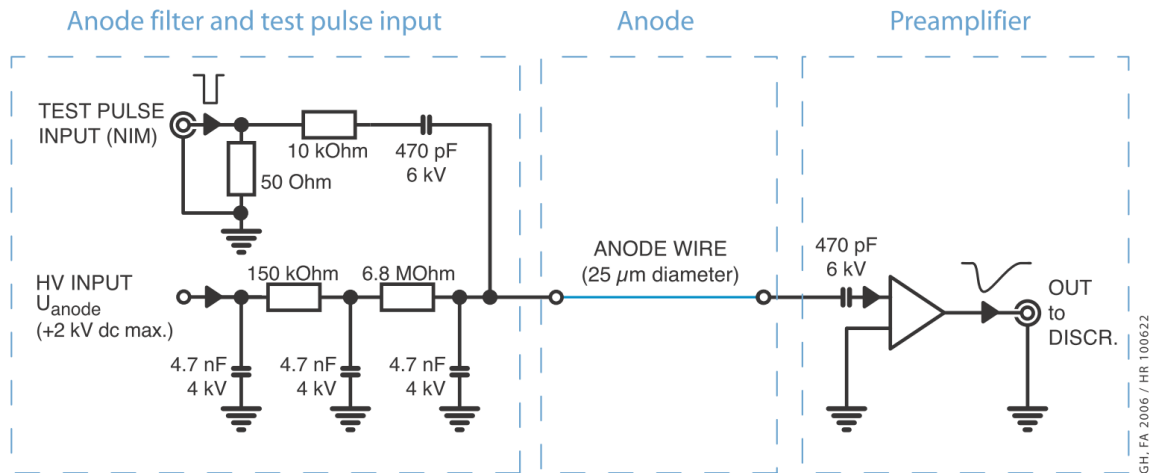


Abbildung 3.7.: Schaltbild[4] der Anodenhochspannung. Links ist der eigentliche Anodenfilter zu sehen, der die Anodenhochspannung glättet. Weiterhin ist die Einspeisung des Testpulses auf den Anodendraht zu sehen. Rechts im Bild ist die Einkopplung des Anodensignals in den Vorverstärker gezeigt.

des Anodendrahtes bis zum Ausgang des Vorverstärkers testen lässt. Gleichzeitig kann damit auch der weitere Signalverlauf getestet werden.



Abbildung 3.8.: Hochspannungsrahmen CAEN SY1527 mit den Hochspannungsnetzteilen für Kathoden (CAEN A1526N), Anoden (CAEN A1821P) und SiPMs (CAEN A1510) (v.l.n.r.)

### 3.2.2. Versorgungseinheiten

#### 3.2.2.1. Hochspannungen

Die einstellbaren Hochspannungen werden vom Hochspannungsrahmen<sup>18</sup> und den verbauten Netzteilen (Abb. 3.8) geliefert. Die maximale Betriebsspannung für das Driftfeld beträgt  $-15\text{ kV}$ <sup>19</sup>. Bei normaler Betriebsspannung von  $-14\text{ kV}$  herrscht im Inneren eine Feldstärke von  $212\text{ V/mm}$ . Die Anodenspannung<sup>20</sup> liegt für die verschiedenen Kammern zwischen  $1850\text{ V}$  und  $2000\text{ V}$ .

#### 3.2.2.2. Niederspannungen

Ebenfalls einstellbar sind die Niederspannungen<sup>21</sup> für die 12 verbauten SiPMs. Vor dem paarweisen Einbau der SiPMs in die Triggerboxen wurden Testmessungen zur Bestimmung des Arbeitspunktes gemacht. Die Paare von SiPMs wurden so gewählt, dass die Differenz der bestimmten Arbeitspunkte möglichst klein ist. Da die Arbeitsbereiche der SiPMs nur eine Breite von  $\sim 0,5\text{ V}$  haben, müssen sich die Spannungen für jeden SiPM einzeln genau ( $\pm 0,02\text{ V}$ ) einstellen lassen. Die Arbeitsspannungen der SiPM liegen zwischen  $69\text{ V}$  und  $71\text{ V}$ .

<sup>18</sup>Hochspannungsrahmen vom Typ CAEN SY1527

<sup>19</sup>Hochspannungsnetzteil vom Typ CAEN A1526N

<sup>20</sup>Hochspannungsnetzteil vom Typ CAEN A1821P

<sup>21</sup>Hochspannungsnetzteil vom Typ CAEN A1510

### 3.2.2.3. Weitere Einheiten

Für Spannungsversorgung, Kabelübergänge und Computeranschlussmöglichkeiten wurden verschiedene VME-Einschübe (Abb. 3.9) gebaut. Der VME-Bus dient nur zur Spannungsversorgung der Einschübe, sofern diese nötig ist.

- Die „VALVE READOUT“ Einschübe stellen die Verbindung zwischen dem PC und den Temperatursensoren her. Dies erfolgt mittels eines USB-OneWire<sup>22</sup>-Adapters. Ganz oben ist der USB-Anschluss für den PC. Die oberen drei Flachbandanschlüsse stellen eine direkte Verbindung zu den Temperatursensoren an den Ventilen der Gasmatrix (siehe 3.2.4.1) her. Der vierte Anschluss ist der Temperaturmasterbus, der die OneWire-Verbindung zu allen anderen Einschüben im Versorgungsrahmen herstellt.
- Die „LV to VDC“ Einschübe versorgen jeweils drei Anodenvorverstärkerplatinen der VDCs mit Spannung. Der Temperaturmasterbus wird benutzt, um die Temperatursensoren auf den Anodenvorverstärkerplatinen mit dem PC zu verbinden.
- Die „PRESSURE LINK“ Einschübe dienen zur Spannungsversorgung von je drei Druckdosen (siehe 3.2.4.2). Ausserdem verbindet der Einschub alle Drucksensoren innerhalb der Druckdosen mit der Analog-Digital-Wandlerkarte im PC. Der unterste Anschluss ist der Temperaturmasterbus, der die Temperatursensoren in den Druckdosen mit dem PC verbindet.
- Die „SUPPLY TR-BOX“ Einschübe versorgen je drei Triggerboxen mit den Versorgungsspannungen für die Verstärker und den Niederspannungen für die SiPMs und stellen eine Verbindung der Temperatursensoren mit dem Temperaturmasterbus her.
- Die Pulser-Einschübe dienen zur Erzeugung von Testpulsen für die verschiedenen Geräte. Derzeit gibt es vier Möglichkeiten, um Testpulse in eine VDC einzuspeisen, nämlich in die LED auf der Triggerbox zur Erzeugung optischer Testpulse, in den Anschlussblock der Anode und in jeden der zwei Vorverstärker für die SiPMs.

### 3.2.3. Triggerlogik

Die Triggerlogik (Abb. 3.10) erzeugt aus den drei Signalen, die eine Kammer liefert, ein zusätzliches Signal. Dies wird aus den beiden diskriminierten SiPM-Signalen per Koinzidenz gebildet und dient als Triggersignal. Als Stoppsignal wird das diskriminierte Anodensignal genutzt. Die Triggersignale aller sechs Kammern werden per OR miteinander verknüpft und an den TRIGGER-Eingang des TDC<sup>23</sup> geliefert. Weiterhin erhält der TDC alle Signale (SiPM1, SiPM2, Trigger und Anode) an den normalen Kanaleingängen. Im Falle eines Ereignisses am TRIGGER werden alle Ereignisse an allen Kanäle erfasst, die innerhalb des eingestellten Triggerfensters des TDCs

<sup>22</sup>Ein-Draht-Bussystem für Sensoren und anderes der Firma Dallas

<sup>23</sup>Time-to-Digital-Converter vom Typ CAEN V767B

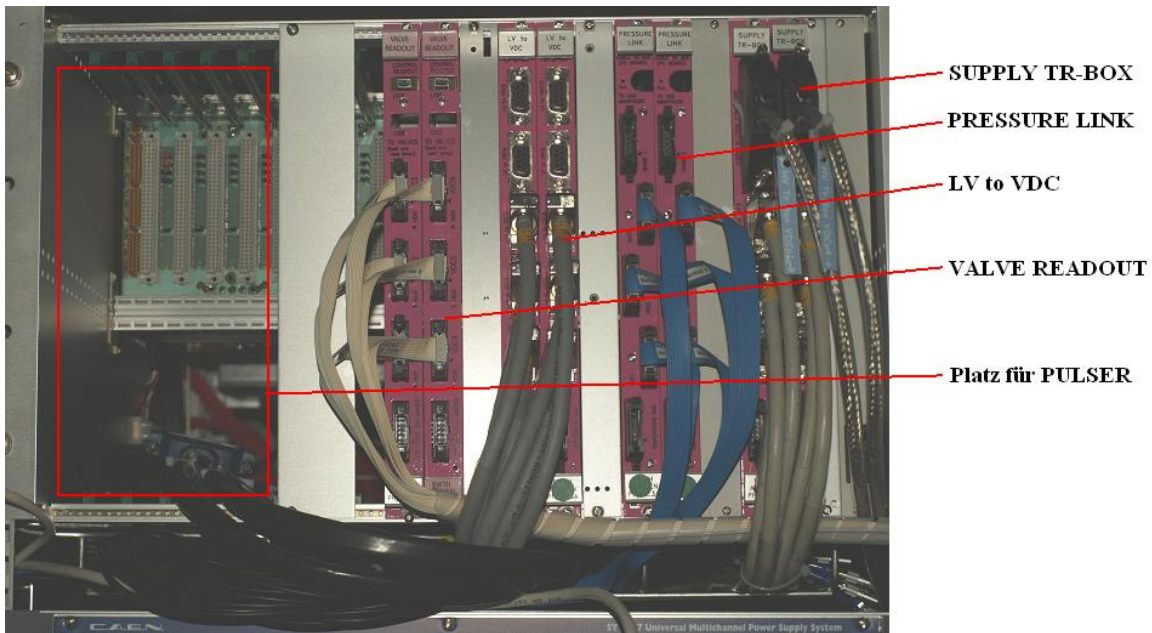


Abbildung 3.9.: Versorgungsraster mit verschiedenen Einschüben. Von links nach rechts:  $2 \times$  VALVE READOUT,  $2 \times$  LV to VDC,  $2 \times$  PRESSURE LINK und  $2 \times$  SUPPLY TR-BOX. Der Platz auf der linken Seite ist für die Pulser-Einschübe vorgesehen, mit denen die Testpulse erzeugt werden.

liegen. Die relativen Zeitpunkte der Ereignisse werden als Event im Ausgabespeicher des TDC abgelegt. Die vier Kanäle, die zur Kammer mit dem Ereignis gehören, müssen alle mit einem gültigen Zeiteintrag im Event auftauchen, sonst wird es nicht zur Bestimmung der Driftgeschwindigkeit verwendet. Alle anderen Kanäle können Zeiteinträge enthalten, da das OR, so wie es hier verwendet wird, einen Randomtrigger für die anderen fünf Kammern darstellt. Aus den Daten des Randomtriggers lassen sich die Rauschraten der einzelnen Kanäle bestimmen.

Da es zum Zeitpunkt des Projektbeginns keine VME-Koinzidenz-Module mit eingebauten Zählern und Kontrolle aller Betriebsparameter per Hand und per Programm zu kaufen gab, wurden eigene Module für die Triggerlogik entwickelt. Dabei handelt es sich um einen Diskriminator (DISCR1x2) mit eingebauten Zählern und Kontrolle über alle Betriebsparameter per Hand und per Programm, analog dazu zwei verschiedene Koinzidenzmodule (COINC3x2 und COINC8x1). Das Modul mit der höchsten Packungsdichte in der Schaltung (COINC3x2) wurde bereits als Prototyp gebaut. Da die Zählfunktionalität der neuen, aber noch nicht verfügbaren Einschübe dringend gebraucht wird, wird diese vorübergehend durch zwei Zählermodule<sup>24</sup> zur Verfügung gestellt.

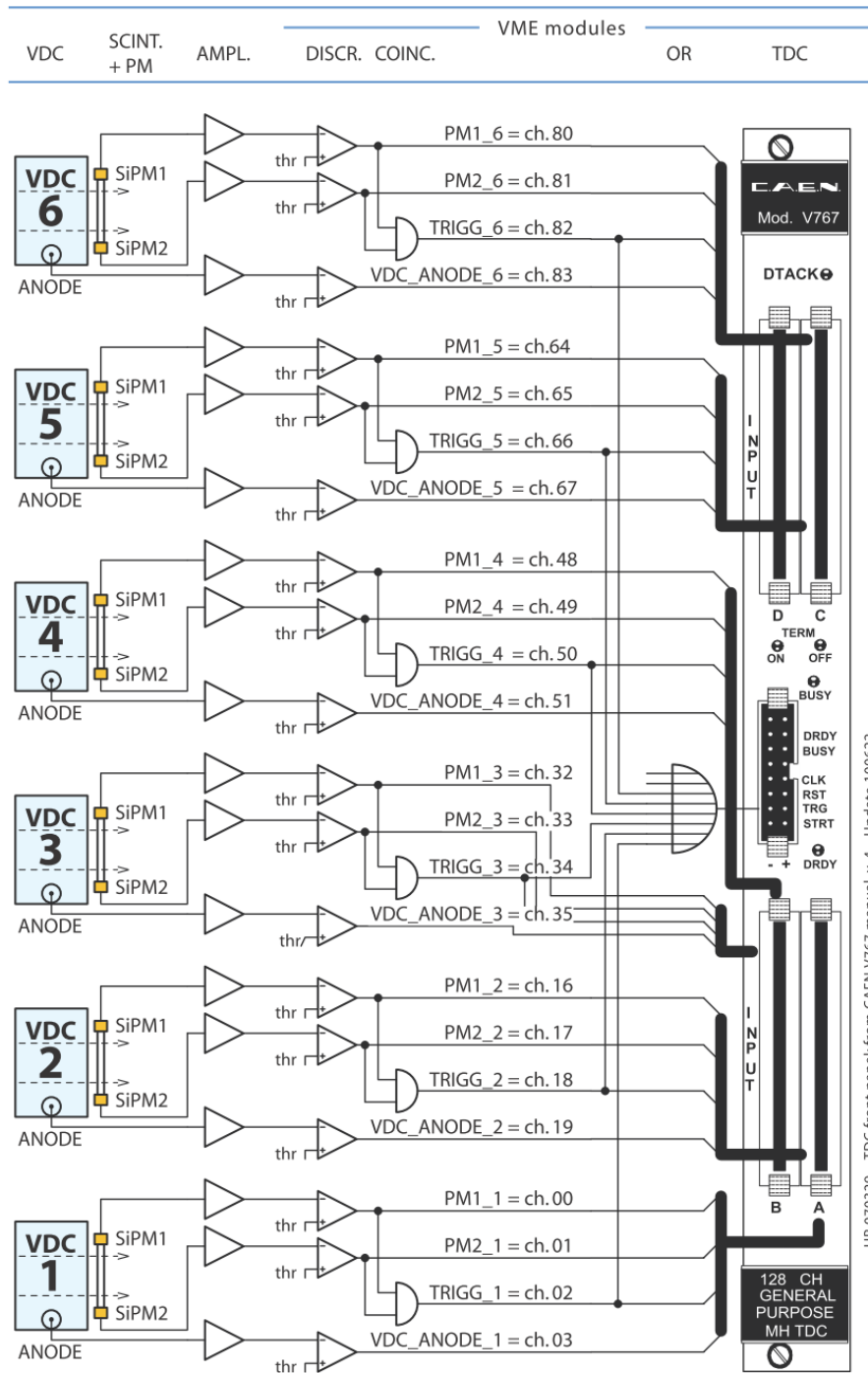


Abbildung 3.10.: Trigger- und Digitalisierungslogik[4] für 6 VDCs an einem TDC

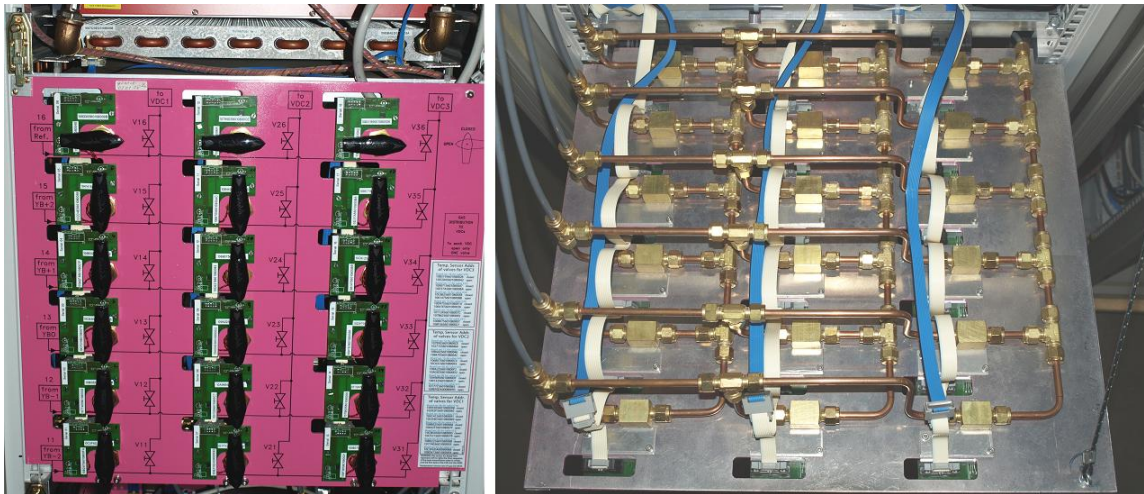


Abbildung 3.11.: Vorder- und Rückansicht der halben Gasmatrix, die andere Hälfte ist baugleich bis auf die Beschriftung. Die von Hand zu bedienenden Ventile ermöglichen es, die Verbindungen innerhalb der Gasmatrix frei zu definieren. Rechts erkennt man die querverlaufenden Rohre der Quellen und die senkrechten Rohre zu den Kammern. Die Flachbandkabel verbinden die Heizwiderstände und Temperatursensoren mit dem „VALVE READOUT“, um die Auslese der Ventilpositionen zu ermöglichen.

### 3.2.4. Überwachungs- und Steuerungsgeräte

#### 3.2.4.1. Gasmatrix

Damit das ankommende Gas von jeder der sechs Quellen (5 Räder + Referenzgas) in einer beliebigen Kammer (oder in mehreren Kammern) vermessen werden kann, wurde eine „Gasmatrix“ (Abb. 3.11) aus 36 von Hand zu bedienenden Ventilen gebaut. Diese erlaubt es, jede Gasquelle mit einer oder auch mehreren frei wählbaren Kammern zu verbinden. Ein schnelles Umschalten zwischen den Kammern ist damit ohne Gasverunreinigung möglich. Es könnte auch das Gas von mehreren Gasquellen gleichzeitig an eine Kammer geleitet werden, das ist aber messtechnisch nicht sinnvoll. Tritt dies auf, so warnt das Ausleseprogramm. Damit per Programm nachvollziehbar ist, welche Gasquelle untersucht wird, sind in die Ventilknebel Heizwiderstände eingebaut, die nach kurzer Aufheizzeit an den in offener und geschlossener Position eingebauten Temperatursensoren eine deutlich messbare Änderung der Temperatur hervorrufen. So wird der aktuelle Ventilzustand erkannt. Die „VALVE READOUT“-Einschübe ermöglichen neben der Auslese der Temperatursensoren auch das Heizen der Ventilknebel. Jeder der drei oberen Flachbandanschlüsse bedient eine Spalte der Gasmatrix.

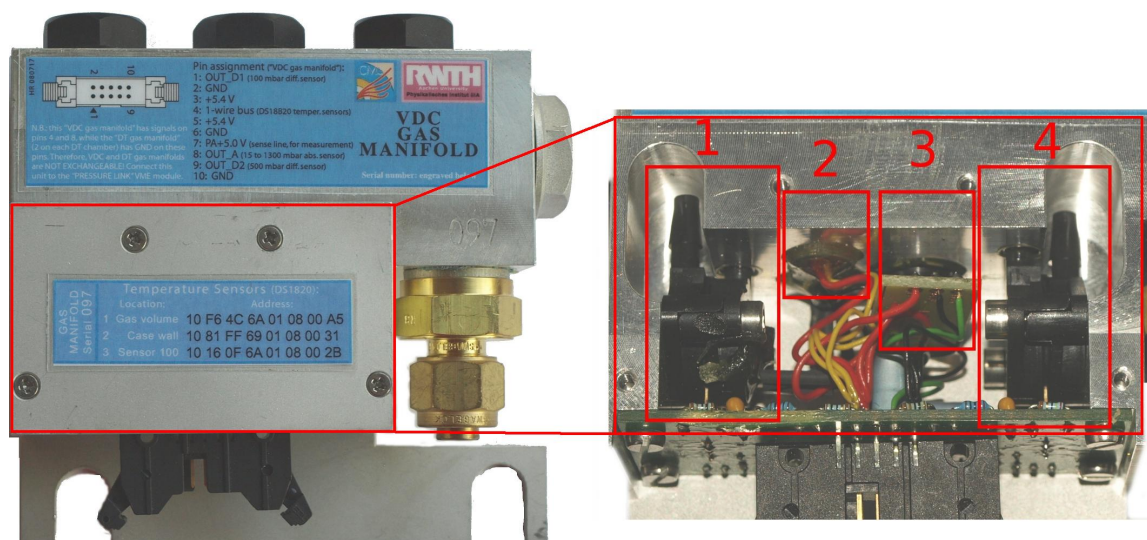


Abbildung 3.12.: Links geschlossene Druckdose, die als Gasverteiler (vier Gasanschlüsse) mit eingebauten Drucksensoren für die großen Driftkammern entwickelt wurden. Unter der Abdeckplatte des angeschlossenen Raums befinden sich ein 100 mbar(1) und ein 500 mbar(4) Differenzdrucksensor sowie ein 1150 mbar(3) Absolutdrucksensor, die den Druck im Gasraum messen. Durch drei eingebaute Temperatursensoren, im Gasvolumen(3), an der Gehäusewand und auf dem 100er Differenzdrucksensor kann die Temperaturabhängigkeit in den Druckmessungen berücksichtigt werden.

### 3.2.4.2. Druckdosen

Die „PRESSURE LINK“-Einschübe stellen die Verbindung der sechs Sensoren in der Druckdose<sup>[18]</sup> zum PC her. Die drei Drucksensoren werden mit einer Analog-Digital-Wandlerkarte<sup>25</sup> im PC ausgelesen. Die drei Temperatursensoren, einer im Gasvolumen, einer an der Gehäusewand und einer auf der Platine, werden mit dem Temperaturmasterbus verbunden. Die verbauten Drucksensoren sind zwei Differenzdrucksensoren mit 100 mbar<sup>26</sup> und 500 mbar<sup>27</sup> Messbereich und ein Absolutdrucksensor mit einem Messbereich bis 1150 mbar<sup>28</sup>.

### 3.2.4.3. Druck- und Flussregler

Jede VDC besitzt einen eigenen Druckregler<sup>29</sup> und einen eigenen Flussregler<sup>30</sup> (Abb. 3.13). Dabei handelt sich um eine kommerziell verfügbare Lösung. Die Fluss- und Druckregelung ist notwendig, da das System über eine unregelmäßige Gaszufuhr versorgt wird.

<sup>24</sup>CAEN V560

<sup>25</sup>National Instruments NI-PCI-6229

<sup>26</sup>Typ: Freescale Semiconductors, Inc. MPX5010DP

<sup>27</sup>Typ: Freescale Semiconductors, Inc. MPX5050DP

<sup>28</sup>Typ: Freescale Semiconductors, Inc. MPXA6115AC6U

<sup>29</sup>Bronkhorst EL-PRESS, HI-TEC Modell P-702C-1K1A-RGD-33-Z Multi-Bus DEPC

<sup>30</sup>Bronkhorst EL-FLOW, HI-TEC Modell F-201C-RGD-33-Z Multi-Bus DMFC



Abbildung 3.13.: Druck- und Flussregler an einer Kammer. Der Fluss wird am Eingang, der Druck am Ausgang der Kammern geregelt. Steuerung und Auslese der zwölf Regler erfolgt über einen Datenbus, dessen Kabel und Adapter oberhalb der Regler zu sehen sind.

#### 3.2.4.4. Temperatursensoren

Zahlreiche Temperatursensoren<sup>31</sup> wurden zwecks Temperaturüberwachung an verschiedenen Positionen im VDC-System verbaut. Diese Sensoren erlauben eine einfache Messung der Temperatur von diversen Komponenten im System.

<sup>31</sup>Dallas DS1820



### 3.3. Überwachung

Die Hochspannungen und zugehörigen Ströme für Kathode und Anode werden mit einer Rate von etwa 170 Hz am Netzteil ausgelesen. Aus jeweils 1000 Messwerten werden die Mittelwerte gebildet, diese werden für die Überwachung der Spannungsstabilität langfristig gespeichert. Da die Messgeschwindigkeit nicht für eine Spitzendetektion ausreicht, muss man sich auf die vom Netzgerät gelieferten Meldungen für Overcurrent (Überstrom), Overvoltage (Überspannung) und Undervoltage (Unterspannung) verlassen. Die dabei erfassten Spannungs- und Stromwerte liefern leider keinen Hinweis auf mögliche Spitzen. Die Meldungen der Over / Under States werden ebenfalls gespeichert. Alle Änderungen der Einstellungen am Gerät selbst oder per Konfigurationsdatei werden protokolliert. Die Niederspannungen für die SiPMs werden mit den Hochspannungen zusammen ausgelesen und in derselben Art und Weise verarbeitet und gespeichert.

Die Messung des Gasdrucks erfolgt mittels der Druckdosen und einer Analog-Digital-Wandler-Karte im PC. Die Gasdruck- und Gasflussregelung erfolgt mittels einer kommerziellen Lösung. Die Gasmatrix liefert die Informationen darüber, welche Gasquelle an welche Kammer angeschlossen ist.

Um eventuelle Fehlerquellen schneller finden zu können, werden eine ganze Reihe von Komponenten temperaturmässig überwacht. Zunächst einmal das Gas selber, da die Temperatur Einfluss auf die Driftgeschwindigkeit hat. Die Gastemperatur wird in der Druckdose und in der Nähe des Vorverstärkers innerhalb der VDC gemessen. Weiter werden die Vorverstärker der Triggerboxen und der Anodenplatinen sowie die Druckdosen überwacht, da Halbleiter im Allgemeinen eine deutliche Temperaturabhängigkeit zeigen. Zusätzlich werden die Zu- und Abläufe der Schrankkühlung überwacht, um Hinweise auf defekte (überhitzte oder ausgefallene) Geräte zu erhalten. Werden Temperaturen gemessen, die außerhalb der konfigurierten Grenzen liegen, so wird dies protokolliert.

## 4. Datennahme- und Steuersoftware

In diesem Kapitel werden die von mir erstellten Programme für die Spannungsüberwachung, TDC-Auslese, Zählerauslese und die Temperatursensoren beschrieben und erklärt.

Die Software für das VDC-System ist sehr modular aufgebaut. Jede Aufgabe, die selbstständig laufen kann, ist als eigenes Modul realisiert und läuft als eigenständiger Prozess. Ist Kommunikation zwischen Prozessen nötig, so läuft diese über Dateien ab. Dies hat den Vorteil, dass ein hängender Prozess keine anderen Prozesse blockieren kann und da es sich bei den Dateien um normal lesbare Textdateien handelt, kann der Benutzer jederzeit sehen, was gerade passiert und was eventuell die Ursache einer Störung ist.

### 4.1. Hoch- und Niederspannungen

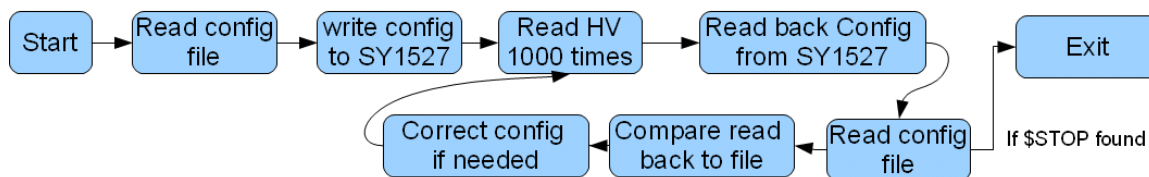


Abbildung 4.1.: Flussdiagramm der HV-Auslese.

Die Hochspannungsüberwachung (Abb. 4.1) ist ein Modul, das auf einer Shared Library von CAEN für den SY1527 Rahmen aufbaut. Diese Library stellt Funktionen zum Lesen und Schreiben von allen Parametern über eine Netzwerkverbindung<sup>32</sup> bereit. Mit Hilfe dieser grundlegenden Funktionen wurden zwei höhere Funktionen erstellt.

Die erste Funktion dient zum Aufzeichnen aller verwendeten Spannungen und Ströme sowie zum Protokollieren von Over- und Understates und Trips. Bei den Over- und Understates handelt es sich um Meldungen der sehr schnellen Spitzendetektion, dass eine Spannung zu hoch oder zu niedrig oder ein Strom zu hoch ist. Die Auslöseschwelle für Spannungsschwankungen liegt bei ungefähr 500 V für die Kathode<sup>33</sup> und bei 90 V für die Anode<sup>34</sup>. Die Meldung für den Strom wird ausgelöst, sobald die eingestellte Stromgrenze überschritten wird. Bleibt der Grund für die Meldung länger bestehen als die eingestellte Tripzeit, wird ein Trip ausgelöst. Dieser hat zur Folge, dass die Spannung für diesen Kanal je nach Einstellung heruntergefahren oder abrupt abgeschaltet wird. Alle Spannungs- und Stromwerte werden über jeweils 1000 Messungen gemittelt

<sup>32</sup>100Mbit Ethernet

<sup>33</sup>3% vom Maximalwert 15 kV, durch die Hardware vorgegeben

<sup>34</sup>3% vom Maximalwert 3 kV, durch die Hardware vorgegeben

und diese Werte gespeichert. Bei der Kathode werden zusätzlich die Minimal- und Maximalwerte für Strom und Spannung aus den 1000 Messungen festgehalten.

Die zweite Funktion ist die Kontroll- und Korrekturfunktion für die Einstellungen aller Betriebsparameter. Nach jedem Aufruf der Aufzeichnungsfunktion werden alle Parameterwerte ausgelesen und mit den Sollwerten und den zuletzt gelesenen Parameterwerten verglichen. Im vollautomatischen Modus werden Abweichungen der Parameter von den Sollwerten protokolliert und die Parameter wieder auf die Sollwerte gesetzt. Im halbautomatischen Modus werden die Änderungen der Einstellungen nur aufgezeichnet, aber nicht korrigiert, daher auch der Vergleich mit den zuvor gelesenen Parametern, um auch erneute Änderungen zu erfassen.

Der halbautomatische Modus ist für den Testbetrieb gedacht, wo Änderungen auf die Schnelle am Einschub und nicht am PC vorgenommen werden. Beim Start im halbautomatischen Modus werden die Werte im Einschub nicht überschrieben, es wird lediglich die Überwachung gestartet. Der vollautomatische Modus dient in erster Linie für den Dauerbetrieb und bezieht seine Werte für die Parameter aus einer Konfigurationsdatei. Ein Wechsel zwischen diesen beiden Betriebsmodi ist jederzeit möglich, da dazu nur ein Eintrag in der Konfigurationsdatei geändert werden muss.

Für die graphische Auswertung steht ein ROOT[19]-Skript zur Verfügung mit einer kleinen Konfigurationsdatei, mittels der sich der auszuwertende Zeitraum und auch zusätzliche Markierungen („Userevents“) angeben lassen. Alle erzeugten Graphen (Abb. 7.1 ff.) sind Zeitdarstellungen, d.h. die x-Achse ist eine Zeitachse. Die vom Benutzer vorgegebenen Userevents, die im Auswertezentrum liegen, erscheinen als senkrechte Linien an den entsprechenden zeitlichen Positionen und enthalten die angegebenen Texte.

## 4.2. VME-Zugriffe

Die Verbindung zu den VME-Modulen (TDC, Zähler, Koinzidenzen) erfolgt über einen USB-VME-Einschub<sup>35</sup> und die zugehörige Programmbibliothek. Diese stellt die Basiszugriffe auf den VME-Bus zur Verfügung. Darauf basierend wurden zwei zunächst unabhängige Module für die TDC- und die Zählerauslese erstellt. Da beide Module auf dasselbe USB-Gerät zugreifen müssen, aber der VME-Zugriff nur einem Modul zur Verfügung steht, wurde in dem TDC-Modul die Möglichkeit implementiert, Zugriffe für andere Programme auszuführen. So kann das Zählermodul trotz laufender TDC-Auslese auf den VME-Bus zugreifen. Die benötigte Zeit für den Zugriff ist jedoch etwas länger als sonst.

### 4.2.1. TDC-Auslese

Das TDC-Modul (Abb. 4.2) überwacht die Einstellungen des TDCs<sup>36</sup> und liest die Messdaten aus. Die gewünschten Einstellungen des TDC werden der zugehörigen Konfigurationsdatei

<sup>35</sup>Wiener VM-USB

<sup>36</sup>CAEn V767B mit 128 Kanälen und einer Zeitauflösung von 0,8 ns

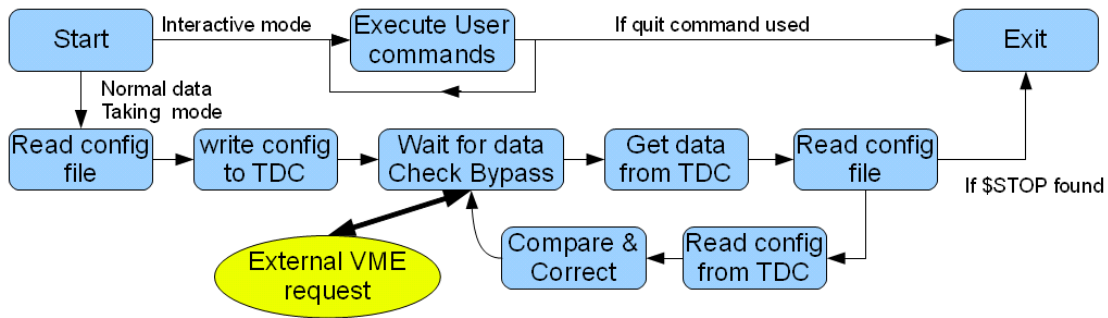


Abbildung 4.2.: Flussdiagramm der TDC-Auslese. Im „Wait for data“ werden auch die VME-Anfragen von anderen Prozessen bearbeitet.

entnommen. Um bei hohen Triggerraten den unnötigen Datenverkehr auf dem VME-Bus zu reduzieren, werden die Daten im Blocktransfermodus übertragen. Das Modul bietet zusätzlich die Möglichkeit eines kurzen TDC-Tests, dazu werden alle wichtigen Parameter auf verschiedene Werte gesetzt und zurückgelesen. Nur wenn alle Werte fehlerfrei zurückgelesen wurden, gilt der Test als bestanden und stellt damit sicher, dass die Kommunikation einwandfrei funktioniert. Weiterhin gibt es einen manuellen interaktiven Modus, in dem alle VME-Befehle von Hand ausgelöst werden können. Dieser dient insbesondere dazu, neue Betriebsparameter zu testen und kurze Tests mit Daten am TDC durchzuführen.

#### 4.2.2. Zählerauslese

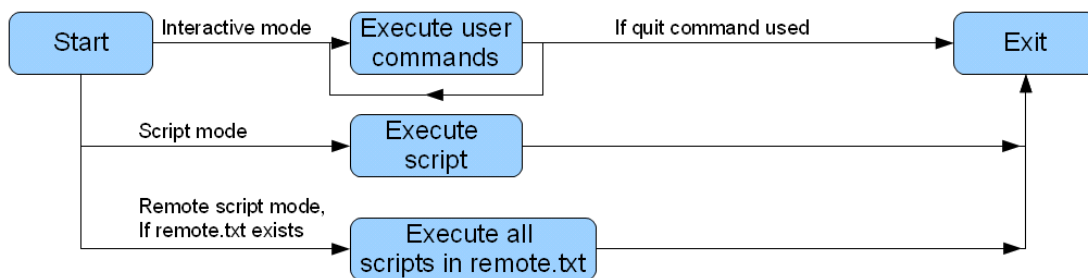


Abbildung 4.3.: Flussdiagramm der Zählerauslese.

Das Zählermodul (Abb. 4.3) dient zur Steuerung der zwei verwendeten Zählereinschübe<sup>37</sup>. Dabei handelt es sich um Zähler mit 16 Kanälen mit jeweils 32 Bit breiten Zählregistern. Diese dienen als temporärer Ersatz für die noch nicht vorhandenen neuen Koinzidenzen und Diskriminatoren mit eingebauten Zählern. Aufgabe der Zähler ist es, Ratenmessungen an allen wichtigen Signalen zu ermöglichen. Daher erhalten die Zähler von allen Signalen eine Kopie, welche aus den zweiten Ausgängen der Diskriminatoren bzw. Koinzidenzen gespeist wird. Das Zählermodul stellt zwei verschiedene Modi zur Verfügung. Zum einen einen manuellen Modus, der die direkte Interaktion

<sup>37</sup>CAEN V560

mit den Zählern ermöglicht, sowie einen Skriptmodus, welcher die in einer Skriptdatei angegebenen Befehle ausführt. Verfügbare Befehle sind unter anderem Starten, Stoppen, Zurücksetzen, Lesen, Warten und eine ganze Reihe von verschiedenen Markerbefehlen, die für die spätere Auswertung nützliche zusätzliche Ausgaben in der Logdatei erzeugen, z. B. Runname, Threshold und Watermark. Diese Angaben sorgen dafür, dass die grafische Auswertung mittels ROOT entsprechende Texte auf den Graphen (Abb. 7.6 ff.) platziert. Zum Beispiel erzeugt Threshold eine Zeile unter dem Graphentitel mit dem Wert der Schwelle an den Diskriminatoren. Das Watermark erzeugt einen kleinen Text am Rand für Information über Autor und Erstellungsdatum. Einige dieser Informationen können später aus anderen Log-Dateien automatisch ermittelt werden.

### 4.3. OneWire-Buszugriffe

Ein weiteres USB-Gerät ist der OneWire-Bus-Adapter<sup>38</sup>. Er ermöglicht die Kommunikation mit den verbauten Temperatursensoren. Diese dienen zwei verschiedenen Zwecken, werden aber vom selben OneWire-Modul bedient. Die eine Funktion des OneWire-Modules (Abb. 4.4) ist das

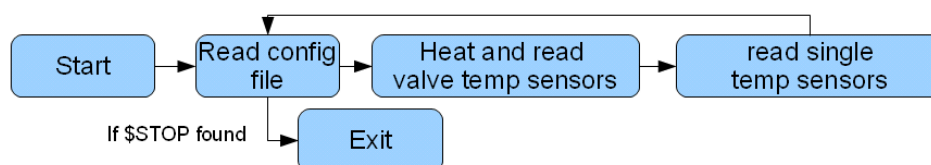


Abbildung 4.4.: Flussdiagramm der Temperaturlauslese.

Auslesen und Aufzeichnen der Werte von etwa 50 Temperatursensoren<sup>39</sup>, die zur Temperaturüberwachung des Gases, der Sensoren, der Triggerboxen und der Schrankkühlung dienen. Die Temperatursensoren liefern bereits digitalisierte Werte und können über einen gemeinsamen Bus ausgelesen werden.

Die zweite Funktion ist das Heizen und Auslesen weiterer 88 Temperatursensoren für die Ventilüberwachung. Dazu gehören die 36 Ventile der Gasmatrix, die 6 Ventile an den Auslässen der Kammern und die 2 Ventile der Referenzgaswahl. Diese Ventile besitzen eine Möglichkeit, die Position des Ventilknebels zu messen. Dazu sind in die Knebel der Ventile Heizwiderstände eingeklebt und in den Positionen „AUF“ und „ZU“ auf einer Platine jeweils ein Temperatursensor platziert. Über mehrere Input-Output-Bausteine<sup>40</sup> für den OneWire-Bus lassen sich die Widerstände heizen. Durch den Vergleich der Temperatur von Auf- und Zu-Sensor lassen sich drei Zustände erfassen, nämlich „AUF“, „ZU“ und „undefiniert“. Weicht der Zustand vom Sollwert ab oder ist undefiniert, so wird dies protokolliert und gemeldet. Die Funktionen werden, wie im Diagramm gezeigt, nacheinander aufgerufen. Diese Modul wird derzeit im Rahmen einer Bachelorarbeit[20] modifiziert und weiterentwickelt.

<sup>38</sup>Dallas DS9490R

<sup>39</sup>Dallas DS1820

<sup>40</sup>Dallas DS2413

## 5. Universaleinschübe

Im Kapitel 3.2.3 wird erwähnt, dass man für die Triggerlogik des VDC-Systems mehrere Diskriminatoren und Koinzidenzen benötigt. Leider gab es zu Projektbeginn keine kommerziell verfügbaren Diskriminatoren und Koinzidenzen, die die gewünschten Anforderungen (vgl. 3.2.3) erfüllten. Auch zum jetzigen Zeitpunkt sind keine dedizierten Module mit diesen oder besseren Eigenschaften erhältlich. Lediglich ein Universalmodul mit einem vom Benutzer programmierbaren FPGA ist jetzt kommerziell verfügbar. Deshalb wurden in der elektrischen Werkstatt des III. Physikalischen Instituts A eigene Diskriminatoren und Koinzidenzen entwickelt.

### 5.1. Anforderungen

Die verlangten Spezifikationen betreffen verschiedenste Eigenschaften der Module. Es muss sich um VME-Einschübe handeln, damit sie in den vorhandenen VME-Rahmen passen und ausgelesen werden können. Sämtliche Betriebsparameter müssen sich, in jedem Modus, immer auslesen lassen. Zusätzlich sollen sie eine weitere, vom VME-Bus unabhängige Schnittstelle besitzen, die Zugang zu den Parametern, z. B. für die Anzeige der Zählerstände, schafft. Alle relevanten Ein- und Ausgänge müssen intern mit Zählern versehen sein, sodass ein externer Zähler für Einzelsignale nur für Kontrollzwecke benutzt werden muss. Die Koinzidenzeingänge sollen mit NIM-Standardpulsen<sup>41</sup> zurecht kommen. Die Diskriminatorschwellen müssen sich mindestens im Bereich von 0,1 V bis 1,0 V einstellen lassen. Die Ausgänge sollen mit in einem weiten Bereich präzise einstellbaren Pulsbreiten ausgerüstet sein. Als Ausgänge sollen vorhanden sein: 1 linear (zeitlich) Out<sup>42</sup>, 2 oder 3 Out, 1 inverted Out ( $\overline{\text{Out}}$ ), 1 Startmarker<sup>43</sup>, 1 Endmarker<sup>44</sup>. Den Zustand des Einschubs muss man auch ohne PC erkennen und bei Bedarf einstellen können, d.h. alle relevanten Parameter müssen von Hand am im Betrieb befindlichen Einschub einstellbar sein und mit LEDs angezeigt werden. Ein- und ausgehende Pulse müssen gut erkennbar angezeigt werden, unabhängig von der Dauer der Pulse. Bei manuellem Betrieb sollen trotzdem die Zustände weiterhin für den PC auslesbar sein. Die Koinzidenz „COINC3x2“ (Abb. 5.1, links) soll zwei Gruppen mit je drei Eingangskanälen beinhalten. Die beiden Gruppen sollen unabhängig voneinander arbeiten und eigene Ausgangsgruppen besitzen. Die Koinzidenz „COINC8x1“

---

<sup>41</sup>NIM Standard Puls: Übergang von 0V nach -0,8 V mit einer Mindestbreite von 1,5 ns. Die meisten Geräte liefern Mindestbreiten von 5 ns

<sup>42</sup>Ausgang entspricht exakt dem gewählten logischen Zustand, ohne zusätzliche Pulsverbreiterung

<sup>43</sup>In der Breite einstellbarer Puls wird generiert, wenn Out an geht

<sup>44</sup>In der Breite einstellbarer Puls wird generiert, wenn Out aus geht

(Abb. 5.1, Mitte) soll eine Gruppe mit acht Eingangskanälen beinhalten. Die einzelnen Eingänge der Koinzidenzen müssen sich unabhängig von einander aktivieren bzw. deaktivieren lassen. Der Betriebsmodus lässt sich zwischen UND und ODER wählen. Der Diskriminator „DISCR1x2“ (Abb. 5.1, rechts) soll zwei unabhängige Eingangskanäle haben, die komplett unabhängig voneinander sind und eigene Ausgangsgruppen besitzen. Die Schwellenwerte sollen sich für jeden Eingangskanal individuell einstellen lassen.

## 5.2. Hardware

Beim Design der Module wurde besonders auf Wiederverwendbarkeit der Subgruppen geachtet. Die wesentlichen Subgruppen sind die drei getrennten Platinen.

- Die in allen drei Modulen verwendete gleiche FPGA-Platine (Abb. 5.2, rechte große Platine) enthält den FPGA und nötige zusätzliche Bausteine für den VME-Bus, den USB-Anschluss sowie einen Spannungswandler, der auch die zusätzlich von der NIM-Platine benötigten 5 V liefert.
- Die NIM-Platine (Abb. 5.2, linke große Platine) ist für alle drei Module unterschiedlich. Aber auch diese drei Platinen verwenden kleinere gemeinsame Subgruppen. So befinden sich dieselben Ausgangsgruppen auf allen drei NIM-Platinen, wenn auch in verschiedenen Anzahlen. Die Koinzidenz „COINC3x2“ enthält, wie auch der Diskriminator, zwei Ausgangsgruppen, die Koinzidenz „COINC8x1“ jedoch nur eine. Für die Eingänge werden auf der Koinzidenz „COINC3x2“ und der Koinzidenz „COINC8x1“ dieselben Baugruppen verwendet. Nur für den Diskriminator „DISCR1x2“ mussten wegen abweichender Eingangseigenschaften eine weitere abweichende Eingangsgruppe entwickelt werden. Auf der NIM-Platine sind alle zeitkritischen und analogen Schaltungen untergebracht.
- Die Frontplatine (Abb. 5.2, links hinter der pinkfarbenen Frontblende) enthält die für die Bedienung durch den Benutzer erforderlichen Schalter, Taster und LEDs. Auch hier wurden Subgruppen mehrfach verwendet. Die dort sichtbaren Lemo-00 Buchsen sind direkt auf der NIM-Platine montiert. Für Wartungsarbeiten lässt sich die Frontplatte mitsamt der Frontplatine direkt abmontieren, ohne Kabel oder dergleichen ablöten zu müssen.

## 5.3. Bedienung

Die Einschübe können über die Schalter und Taster auf den Frontplatte unmittelbar bedient werden. Die Nutzung eines PC ermöglicht aber auch eine mittelbare Einstellung der Betriebsparameter der Einschübe. Da sich die Parameter auslesen lassen, können diese auch am PC dargestellt werden. Die Anzeige der aktuell gültigen Parameter erfolgt über die LEDs an den

VME-Einschüben. Die Zähler- und Presetfunktionen sind nicht direkt am VME-Einschub einstellbar. Diese Funktionen lassen sich nur (via VME oder USB ) mit einem PC bedienen, da es keinen Platz für eine Anzeige auf der Frontplatte gibt.

## 5.4. Firmware

Bei einem FPGA (Field Programmable Gate Array) handelt es sich um einen integrierten Schaltkreis, in dem sich logische Schaltungen programmieren lassen. Übersetzt bedeutet der Name soviel wie „im Anwendungsfeld programmierbare Anordnung von Logikgattern“. Es gibt drei Bauarten von FPGAs. Fuse-FPGAs lassen sich nur einmal konfigurieren, der Flash-Typ erlaubt ca. 100.000 Konfigurationsänderungen, während der SRAM-Typ beliebig oft geändert werden kann. Fuse- und Flash-basierte FPGA sind nach dem Einschalten sofort betriebsbereit, während die SRAM-basierten erst ihre Konfiguration aus dem zusätzlichen externen Speicher laden müssen, was einige Millisekunden in Anspruch nehmen kann. Bei den auf der FPGA-Platine verwendeten FPGAs<sup>45</sup> handelt es sich um SRAM-basierte. Die benötigten Konfigurationsdateien können auf verschiedenen Wegen erstellt werden. Man kann einen Schaltplan zeichnen oder eine Beschreibung mittels einer Hardware Description Language (HDL, Hardwarebeschreibungssprache) wie Verilog oder VHDL anfertigen. Diese Beschreibung, egal, ob als Text oder Schaltplan, wird dann von einem Synthesewerkzeug in eine Netzliste umgesetzt und schließlich in das benötigte Format für den jeweiligen FPGA-Typ gepackt. Diese letzte Datei programmiert den FPGA zur Laufzeit. Zur Programmierung („Beschreibung“) des FPGAs wird im Projekt die Hardwarebeschreibungssprache VHDL verwendet. VHDL steht für Very High Speed Integrated Circuit Hardware Description Language (auch VHSIC Hardware Description Language) und ist eine HDL, mit der sich komplexe Digitalschaltungen leicht beschreiben und realisieren lassen.

### 5.4.1. Programmstruktur

Die drei verschiedenen Designs für die drei Universaleinschübe besitzen alle eine ähnliche Grundstruktur. Es handelt sich dabei um das Grundgerüst einer „Finite-State Machine“ (FSM). Darin wird jeder mögliche Zustand beschrieben und definiert, was passieren soll, wenn dieser Zustand erreicht wird. Das Finite (endlich) bezieht sich nur auf die Anzahl der Zustände, nicht darauf, ob es einen Endzustand und damit ein Anhalten der Maschine gibt. Um die Haupt-FSM nicht zu unübersichtlich werden zu lassen, gibt es ausgegliederte Submodule, die über fest definierte Schnittstellen verfügen und teilweise auch als FSM entworfen wurden. Die Abbildung 5.3 zeigt im Überblick die verschiedenen Submodule und ihre Anbindung an das Hauptmodul.

---

<sup>45</sup>Xilinx Spartan XC3S500E FGG320



## 5.4.2. Ausgegliederte und mehrfach genutzte Submodule

Dabei handelt es sich um die Module für die Zähler, weil diese mehrfach vorkommen und nur bedingt Einfluss auf die Haupt-FSM nehmen, sowie die beiden Kommunikationsmodule für VME und RS232 und den Entpreller für die Taster und Schalter.

### 5.4.2.1. Zähler

Das Zählersubmodul stellt einen Zähler mit einigen speziellen Funktionen zur Verfügung. Um das totzeitfreie Auslesen bei laufendem Betrieb zu gewährleisten, gibt es zwei Zählregister. Liegt kein Auslesesignal an, werden beide Zähler parallel geführt. Liegt jedoch ein Auslesesignal an, wird nur ein Register hochgezählt, während das andere unverändert „eingefroren“ bleibt, bis das Auslesesignal wieder auf 0 geht. Liegt zum Zeitpunkt des Rückgangs kein Zählimpuls an, so wird das zweite Register mit einer Kopie des ersten synchronisiert, anderenfalls wird es durch den aktuellen Zählimpuls aktualisiert. Durch diese Logik werden also alle Zählimpulse gezählt, die Zählung nicht durch die Auslese beeinflusst und der Zähler auch dann richtig aktualisiert, wenn die letzten Zählimpulse während der letzten Auslese eintrafen, so dass der Zähler beim erneuten Auslesen die aktuellen Werte enthält.

Listing 5.1: Funktionaler VHDL-Codeauszug des Zählersubmoduls

```

1 entity async_cnt_reset_read_copy is
2   Generic (
3     cnt_breite : integer := 64
4   );
5   Port ( input : in  STD_LOGIC;
6         reset : in  STD_LOGIC;
7         read_request : in  STD_LOGIC;
8         cnt_enable : in  STD_LOGIC;
9         cnt_out : out  unsigned (cnt_breite-1 downto 0)
10  );
11 end async_cnt_reset_read_copy;
12 architecture Behavioral of async_cnt_reset_read_copy is
13   signal cnt, cnt_copy : unsigned(cnt_breite-1 downto 0);
14   signal copied: STD_LOGIC:= '0';
15 begin
16   process (input, reset, read_request, cnt, copied)
17   begin
18     if reset='1' then
19       cnt <= (others => '0');
20       cnt_copy <=(others => '0');
21     elsif input'event and input = '1' then
22       if cnt_enable='1' then
23         cnt <= cnt + 1;

```

```

24     if read_request = '1' then
25         cnt_copy <= cnt_copy;
26         copied <= '0';
27     else
28         cnt_copy <= cnt + 1;
29         copied <= '1';
30     end if;
31     end if; -- cnt_ena
32 end if; -- reset / input
33 -- Kopie anfertigen auch wenn kein weiterer input puls gekommen ist
34 if read_request = '0' and reset = '0' and copied = '0' then
35     cnt_copy <= cnt;
36     copied <= '1';
37 end if ;
38 end process;
39 cnt_out <= cnt_copy;
40 end Behavioral;

```

Der Auszug 5.1 beinhaltet die funktionsrelevanten Teile des Codes, die vollständige Version ist im Anhang A.1 zu finden. Die Zeilen 1 – 11 definieren die Schnittstelle für die Benutzung im Hauptmodul. Dazu werden hier die Signale definiert. Das zu zählende Signal wird in „input“ geleitet. Das Signal „reset“ dient zum Zurücksetzen des Zählers, „cnt\_enable“ definiert, ob der Zähler zählt oder nicht und „read\_request“ sperrt die Kopie, sodass sie fehlerfrei ausgelesen werden kann. Das Signal „cnt\_out“ enthält den Zählerstand. Über „cnt\_breite“ kann die Breite der Zählregister verändert werden, standardmäßig beträgt sie 64 Bit. Ab Zeile 12 folgt das Verhalten (Logik) des Submoduls. Zunächst werden die 2 Zählerregister („cnt“ und „cnt\_copy“) in der in „cnt\_breite“ definierten Größe innerhalb des Submoduls deklariert. Der Befehl „process“ definiert, bei welchen Signalen er ausgeführt wird, in diesem Fall „input“ , „reset“, „read\_request“, „cnt“ und „copied“. Ändert sich eines dieser Signale, so wird „process“ ausgeführt. Die erste „if“-Abfrage (Zeile 18) prüft, ob es einen Reset gibt und setzt in diesem Fall die beiden Zählregister auf 0. Wenn es keinen Reset gab, wird mit der zweiten „if“-Abfrage (Zeile 21) geprüft, ob es eine steigende Flanke im „input“ gibt, d.h. ob ein Zählpuls angekommen ist. Bevor nun das Zählregister erhöht wird, wird mit dem „if“ in Zeile 22 geprüft, ob gezählt werden soll und dementsprechend „cnt“ erhöht oder nicht. Wenn gezählt wird, wird danach geprüft, ob „read\_request“ gesetzt ist. Wenn ja, wird „cnt\_copy“ nicht erhöht und „copied“ auf 0 gesetzt. Ist „read\_request“ nicht gesetzt, wird „cnt\_copy“ mit dem um eins erhöhten „cnt“-Wert gefüllt und „copied“ auf 1 gesetzt. Die Erhöhung um eins ist nötig, da die Signale erst nach dem Beenden des „process“ aktualisiert werden. Die „if“-Abfrage in Zeile 34 stellt sicher, dass die Kopie des Zählerregisters aktualisiert wird, wenn der „read\_request“ wieder auf 0 und eine Aktualisierung nötig („copied“ auf 0 ) ist. Zeile 39 verknüpft schließlich die Kopie des Zählerregisters mit dem Ausgangssignal „cnt\_out“, welches vom Hauptmodul weiterverwendet wird.

#### 5.4.2.2. RS232

Der für den Benutzer zugängliche USB-Anschluss ist der Anschluss eines USB-RS232 Adapters. Der FPGA sieht nur die RS232-Anschlüsse dieses Bausteins, daher wird auf dem FPGA eine RS232 Schnittstelle implementiert. Dieses Submodul untergliedert sich in drei Dateien, das Interface (Anh. A.3), welches in der Haupt-FSM eingebunden wird, sowie einen Sender (Anh. A.5) und einen Empfänger (Anh. A.4). Das Interface stellt die Verbindung zwischen dem Hauptmodul und den Submodulen des RS232-Moduls her. Das Sender-Modul schreibt den übergebenen Registerwert als serielles Bitmuster auf die RS232-Schnittstelle. Das Empfänger-Modul liest die ankommenden seriellen Bitmuster und stellt sie als Register zur Verfügung.

#### 5.4.2.3. VME / VME IRQ

Das VME-Submodul wurde entsprechend dem VME-Standard[21] geplant. Alle nötigen Signale des VME-Busses wurden definiert. Die Tristate IOs, Bausteine die sowohl als Eingang und auch als Ausgang dienen, wurden passend zugewiesen. Was noch implementiert werden muss, ist die Logik. Der Sourcecode befindet sich im Anhang A.6.

#### 5.4.2.4. Entpreller

Das Submodul Entpreller(Anh. A.2) wird für jeden Taster und Schalter benutzt und unterdrückt die bei mechanischen Kontakten entstehenden ungewollten Nachpulse. Diese würden ein präzises Einstellen von Werten unmöglich machen, da je nach Bauart der Taster und Schalter diese bis zu einigen hundert Mal prellen können. Um diese ungewollten Pulse zu unterdrücken, leitet dieses Submodul die Taster- und Schalterzustände erst dann an das Hauptmodul weiter, wenn der Zustand über eine vorgegebene Zeit konstant ist. Die Zeit beträgt in dieser Version 511 Takte, also 10,22  $\mu$ s.

#### 5.4.3. Performance des Designs

Beim Erstellen der Netzliste für das FPGA-Design wurde aufgrund der maximalen Signalfadlänge mit einer Laufzeit von 16 ns eine maximale Frequenz von 62,5 MHz angegeben, der externe Takt beträgt aber 100 MHz.

Da dieser Pfad über den Programmablauf entscheidet, lässt er sich nicht unterteilen oder in einem Register bis zum nächsten Takt zwischenspeichern. Daher muss die maximale Laufzeit kleiner als die Taktzeit sein. Alle Versuche, diesen Pfad zu verkürzen, schlugen fehl. Sowohl die Vorgabe eines Timing-Constraints von 10 ns für diesen Pfad und einer über Nacht laufenden Optimierungsfunktion führten lediglich zu einer Verbesserung um 0,5 ns. Auch die Versuche, es durch eine andere Programmierung zu lösen, brachten keine brauchbaren Ergebnisse. Das Hauptproblem dieses Pfades liegt darin, dass er durch 19 Gatter läuft, die jeweils eine Verzögerung von

etwa 0,4 ns haben. Daher ergibt sich eine Mindestlaufzeit von 7,6 ns bei optimaler Lage der Gatter. Aufgrund der räumlichen Trennung ist diese aber nicht zu erreichen. Eine Reduktion der Anzahl der durchlaufenen Gatter durch eine andere Programmierung war ebenfalls nicht erfolgreich.

Daher wird jetzt der externe Takt von 100 MHz(10 ns) intern durch einen Frequenzteiler auf 50 MHz(20 ns) reduziert.

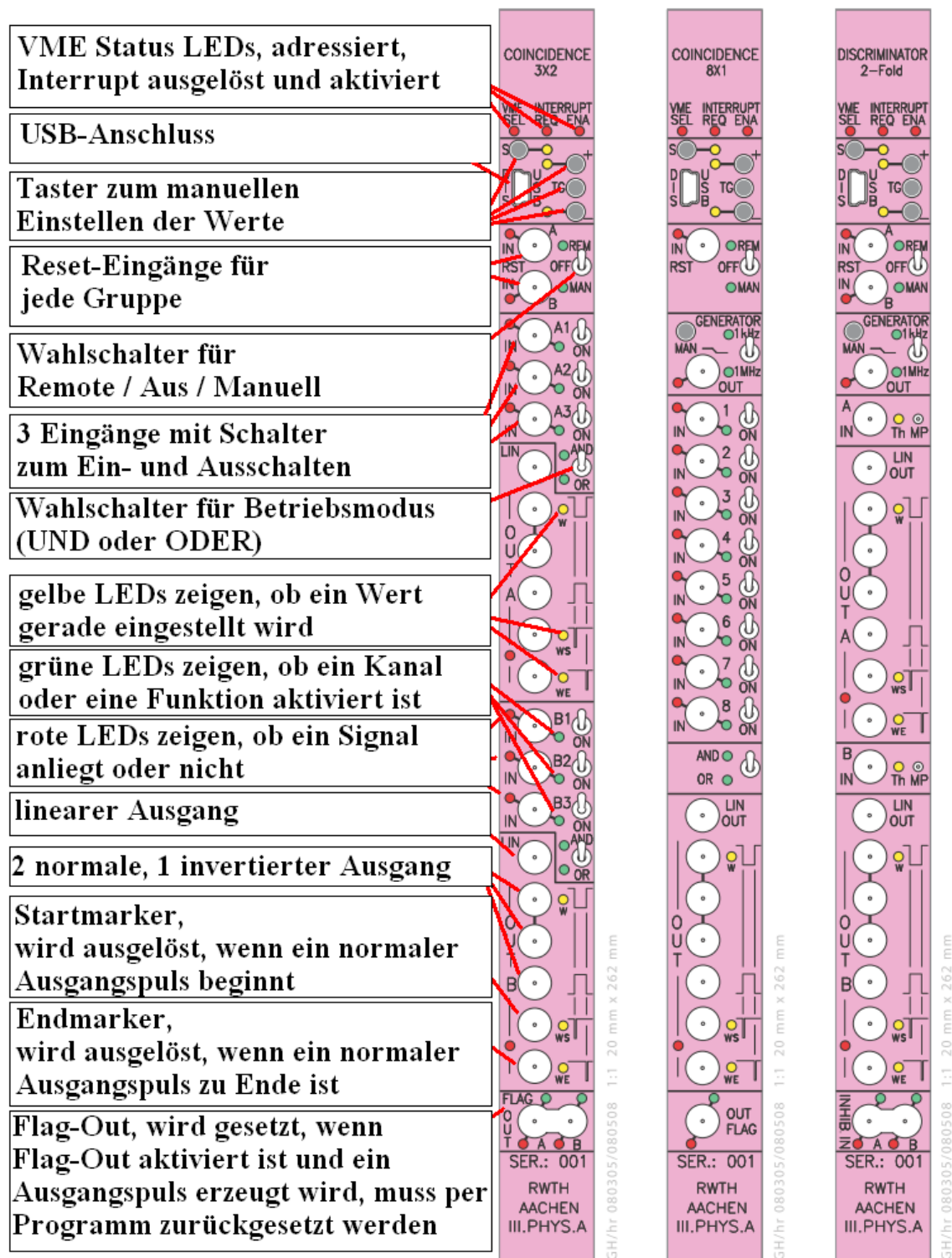


Abbildung 5.1.: Die Abbildung[4] zeigt die drei Frontblenden für die Universalmodule. Links ist die COINC3x2 mit einigen Erläuterungen, in der Mitte die COINC8x1 und rechts der DISCR2x1.

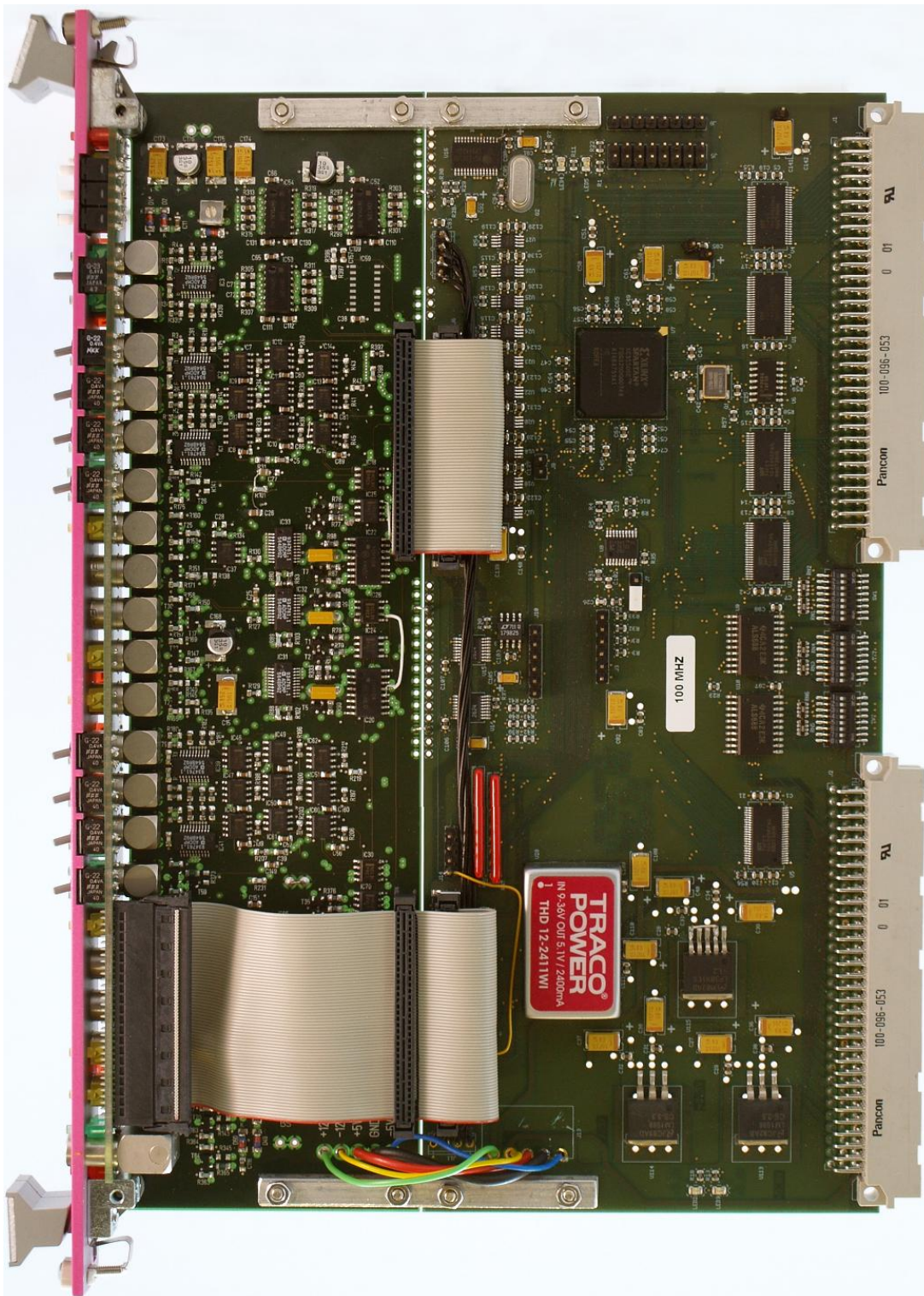


Abbildung 5.2.: Die drei Platinen des Prototyps des VME-Einschubs Koinzidenz „COINC3x2“, links hinter der pinkfarbenen Frontplatte befindet sich die schmale Frontplatte. Die linke große Platine ist die NIM-Platine und die rechte ist die FPGA-Platine. Die NIM- und die FPGA-Platine werden mittels der beiden Metallbrücken oben und unten fest miteinander verschraubt. Für die Serienfertigung können FPGA- und NIM-Gruppe auf einer Platine zusammengefasst werden.

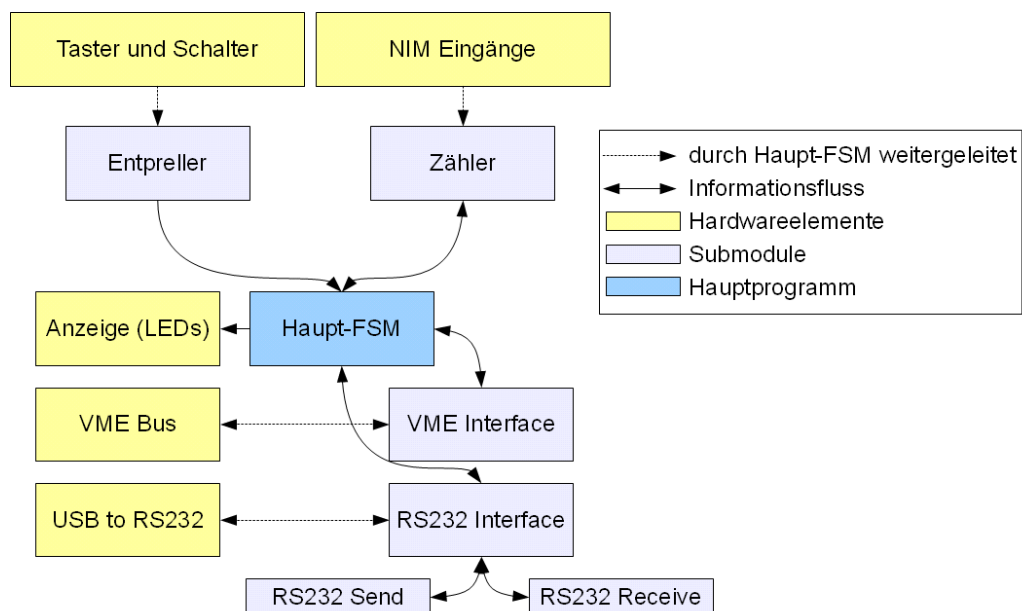


Abbildung 5.3.: Übersichtsbild des Hauptmoduls. Es zeigt die Kommunikationsrichtungen der Submodule und Hardwareelemente

## 6. Testmöglichkeiten

Zum Testen der Funktionstüchtigkeit der Universaleinschübe gibt es eine ganze Reihe von Möglichkeiten, die ohne spezielle Hardware, wie z.B. einem Logic Analyzer, auskommen. Dies ist für die Diagnose der installierten Geräte, am Einsatzort, mit einfachen Mitteln besonders hilfreich. Die nachfolgenden Tests stellen sicher, dass die verwendeten Einschübe die Funktionalität gewährleisten, die von ihnen erwartet wird. Bis auf die Tests mit den Zählern wurden diese Tests auch mit den aktuell benutzten Modulen durchgeführt. Ohne diese Verifikation der korrekten Arbeitsweise hätten die Messungen keine Aussagekraft. Auch der Prototyp der Koinzidenz „COINC3x2“ wurde auf diese Art und Weise erfolgreich getestet.

### 6.1. Koinzidenztests

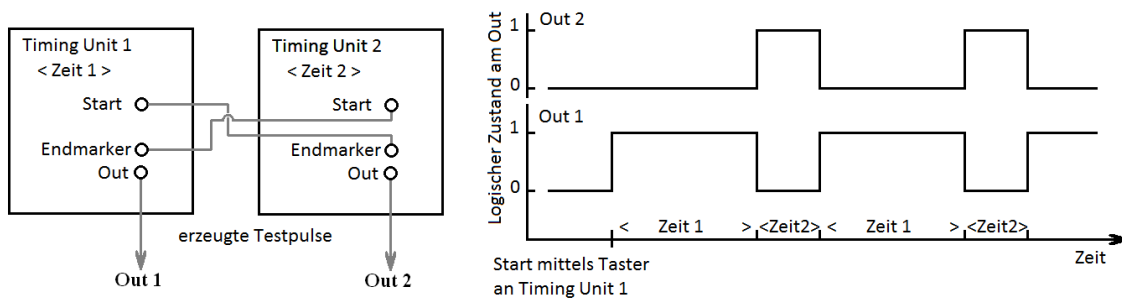


Abbildung 6.1.: Links ist die Verkabelung der „Timing Units“ zum Erzeugen eines periodischen Signals zu sehen. Rechts sind die zwei an den Ausgängen der Timing Units erzeugten logischen Signalverläufe dargestellt

Zum Testen der grundlegenden Funktionen reichen ein Fan-Out<sup>46</sup> und zwei Timing-Units. Damit lassen sich die für den UND- bzw. ODER-Test benötigten Signale erzeugen. Für den UND-Test werden alle aktivierten Kanäle mit Signalen aus dem Fan-Out versorgt, wobei man zunächst jeden Kanal einzeln aktiviert und dann erst mehrere. Der Fan-Out wird durch ein periodisches Signal, das durch eine geeignete Kopplung (Abb. 6.1) der Timing-Units erzeugt wird, gespeist. Wenn alles richtig funktioniert, sollte der Ausgang nun „high“ sein.

Dann wird bei jeweils einem Kanal das Kabel abgezogen, der Kanal bleibt aber aktiviert. Wenn das Ausgangssignal aus geht, sind „permanent-high“ Eingänge ausgeschlossen. Dann wird geprüft, ob deaktivierte Kanäle fälschlicherweise Einfluss auf den Ausgang haben. Dazu wird der zu

<sup>46</sup>Signalervielfacher, ein Eingang, mehrere Ausgänge



testende Kanal deaktiviert und das Kabel abgezogen und wieder aufgesteckt. Änderungen am deaktivierten Kanal dürfen keine Änderungen am Ausgang zur Folge haben. Ein weiterer Test prüft, ob die Kanäle schnell genug reagieren. Dazu werden zwei Kanäle aktiviert. Einer wird aus der ersten Timing-Unit gespeist, der andere aus der zweiten. Die zwei Timing-Units erzeugen zwei periodische Signale, die nie gleichzeitig „high“-Pegel aufweisen. Durch die Wahl der Kabellängen zur Signaleinspeisung können die Pulse zeitlich so variiert werden, dass sie sich gerade oder gerade nicht überlappen. Im UND-Modus sollte der Ausgang den Überlapp als Koinzidenz anzeigen. Für den ODER-Modus lassen sich die meisten Tests ähnlich durchführen. Getestet werden sollte auch hier zunächst jeder Kanal für sich, sowohl mit als auch ohne Signal, um defekte Kanäle zu erkennen. Tests mit überlappenden Pulsen lassen sich im ODER-Modus nur sinnvoll mit Hilfe eines Oszilloskops durchführen.

## 6.2. Diskriminatortests

Zum Testen der Diskriminatoren benötigt man einen Pulsgenerator mit einstellbarer Pulshöhe sowie ein Multimeter. Am Messpunkt kann die an der Koinzidenz eingestellte Schwelle mit dem Multimeter abgegriffen werden. Stellt man nun am Pulsgenerator eine möglichst knapp unter der Schwelle liegende Pulshöhe ein, sollte am Ausgang des Diskriminators kein Signal ankommen. Eine minimale Erhöhung sollte dann ein Signal liefern. Verwendet man die beiden Kanäle gleichzeitig, kann man die Ausgänge mit einer Koinzidenz im UND-Modus überwachen. Dazu wird ein Kanal auf eine niedrige Schwelle gesetzt, um ein sicheres Auslösen dieses Kanals zu erreichen. Der andere wird, wie zuvor beschrieben, getestet.

## 6.3. Timingtest

Die Einschübe sind zwar so ausgelegt, dass alle Kanäle dieselbe Laufzeit haben sollen, durch Produktionsschwankungen der Bauteile und Unterschiede in der Bestückung (Lötstellen) kann es aber sein, dass dies nicht hundertprozentig erreicht wird. Ist eine bekannte Signallaufzeit wichtig, muss diese gemessen werden. Um die Signallaufzeit einzelner Kanäle zu messen, benötigt man ein Oszilloskop mit der gewünschten Zeitaufösung. Über den Fan-Out wird ein Signal direkt an das Oszilloskop weitergeleitet, während ein zweites Signal das Oszilloskop über den zu messenden Kanal und den zugehörigen Ausgang erreicht. Die Kabellängen vom Fan-Out bis zum Oszilloskop sind so zu wählen, dass beide Signalwege dieselbe Kabellänge beinhalten. Die Laufzeit für den vermessenen Kanal kann direkt auf dem Oszilloskop als Zeitdifferenz zwischen den beiden Signalen abgelesen werden.

## 6.4. Zählertests

Die Verifikation, dass die Zähler richtig arbeiten, lässt sich mit 2 Koinzidenzeinschüben durchführen. Zum Test werden die Eingangszähler mit demselben Signal gespeist und sollten stets den gleichen Stand haben. Für den Test des Ausgangszählers wird der zugehörige Ausgang in den Eingang eines anderen Zählermoduls gespeist. Auch hier müssen die Zählerstände stets gleich sein.

## 6.5. RS232-Test

Da sich der USB-RS232 Adapter im PC als COM-Schnittstelle meldet, kann die Kommunikation mittels der Freeware COM-Terminal [22] oder eines anderen Programms für einfache COM Operationen getestet werden. Dazu werden die Lese- und Schreibbefehle von Hand ausgeführt.

## 6.6. Ferndiagnose

Einige Funktionen sollten sich auch aus der Ferne per PC testen lassen.

- Tote Eingangskanäle lassen sich ausschließen, wenn die zugehörigen Eingangszähler die bekannte typische Rate oder überhaupt eine Rate erfassen (Koinzidenzen).
- Tote Ausgangskanäle lassen sich ausschließen, wenn die Zähler an den Ausgängen eine Rate erfassen (Koinzidenzen und Diskriminator).
- Durch Vergleich der Eingangs- und Ausgangszähler lässt sich erkennen, ob die Hauptfunktion eines Einschubs funktioniert (Koinzidenzen). Beim Umschalten des Modus von UND auf OR sollte die Rate am Ausgang steigen. Deaktiviert man einen Eingangskanal, so sollte die Rate sinken. Ist nur ein Eingang aktiv, sollten Eingangs- und Ausgangszähler gleiche Werte aufweisen.
- Werden ungenutzte Kanäle permanent mit einem bekannten Testsignal gespeist, so können diese Kanäle für die Diagnose aktiviert werden und erlauben eine Fehlersuche mit konstanten Raten.

Durch die dauerhafte Überwachung der Betriebsparameter kann sichergestellt werden, dass die Datennahme nicht durch falsche Einstellungen oder fehlerhafte Module unbemerkt beeinträchtigt wird. Eventuelle Fehler können zeitnah festgestellt werden, womit fragwürdige Messungen erkannt und markiert werden. Eine Diagnose erlaubt die Einkreisung der möglicherweise fehlerhaften Komponenten und somit eine zügige Instandsetzung.

## 7. Ergebnisse der Messungen

Mit jedem der im Laufe der Zeit entstandenen Softwaremodule eröffneten sich neue Möglichkeiten für Messungen. Die in diesem Kapitel gezeigten Messung stellen die ersten Ergebnisse von typischen Messungen dar. Weitere detaillierte systematische Messungen sind im Rahmen nachfolgender Arbeiten vorgesehen.

### 7.1. Hochspannungsmessungen

Mit dem Beginn der Hochspannungsüberwachung wurden die ersten Langzeitmessungen zur Hochspannungsstabilität der VDCs gestartet. Die Kammern halten die Hochspannung im Allgemeinen problemlos. Sporadisch auftretende Störungen wurden systematisch auf ihre Ursache hin untersucht, lokalisiert und konnten durch Korrekturmaßnahmen an allen Kammern behoben werden. Die jeweiligen Korrekturmaßnahmen behoben die Fehler dauerhaft. Bei einer „gesunden“ Kammer ist die Hochspannung und der Strom sehr stabil. In den Abbildungen 7.1 und 7.2 sind Spannung und Strom einer gesunden Kammer gezeigt. Nur bei sehr wenigen Messungen gibt es eine Abweichung von ein paar Volt. Während des gesamten dargestellten Zeitraums liegt die Spannung zwischen 14000 V und 14005 V. Auch der Strom ist nach einem Sprung von  $25,6 \mu\text{A}$  auf  $25,7 \mu\text{A}$  im Rahmen der Messgenauigkeit konstant. Die Abb. 7.3 und 7.4 zeigen eine „kranke Kammer“. Die Kathodenspannung bricht immer wieder kurzzeitig aufgrund von Entladungen ein und sinkt damit kurzfristig von 14000 V auf bis zu ca. 13600 V. Auch der Strom ist nicht mehr konstant, sondern zeigt starke Schwankungen von  $24 \mu\text{A}$  bis  $36 \mu\text{A}$ . Solche Messungen wurden zum Anlass genommen, nach den Ursachen zu suchen. Dabei wurden Schwachstellen festgestellt und durch konstruktive Änderungen, z. B. dem Einbau zusätzlicher Kunststoffteile, behoben. Während bei einer „gesunden“ Kammer keine Over-/Understates, also zu große Spannungsabweichung oder zu hoher Strom, auftreten, zeigen sich bei einer „kranken“ Kammer gelegentliche Over-/Understates (Abb. 7.5). Treten diese im Abstand von wenigen Sekunden auf, so handelt es sich um eine „defekte“ Kammer. Eine solche Kammer eignet sich auf Grund des variierenden elektrischen Feldes nicht mehr für hochpräzise Messungen der Driftgeschwindigkeit. Eine Unterscheidung zwischen Over- und Understates findet in der Auswertung nicht statt. Die allgemeine Hochspannungsfestigkeit der Kammern hat ein für den Dauereinsatz taugliches Niveau erreicht, denn die nominale Kathodenspannung  $-14 \text{ kV}$  wurde für mehr als 6 Monate ohne Störung gehalten.

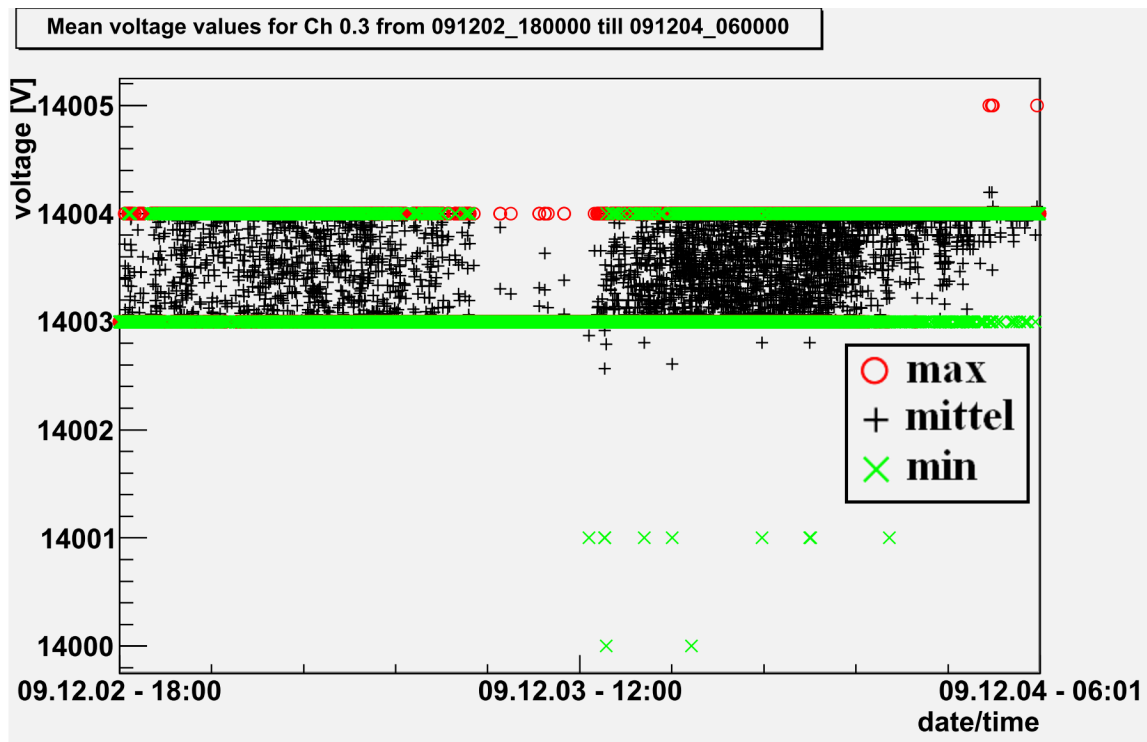


Abbildung 7.1.: Zwei-Tages-Ausschnitt aus dem Graph der Hochspannung einer Kammer ohne Auffälligkeiten. Die Schwankungen der Spannung betragen nur etwa 1 V. Grüne Kreuze (×) stellen den Minimalwert, rote Kreise (○) den Maximalwert und schwarze Plus-Symbole (+) den Mittelwert von 1000 Messungen dar. (Kathodenspannung VDC4; ca. 20000 Messpunkte, Digitalisierungsschritt 1 V)

## 7.2. Ratenmessungen

Mit den externen Zählern konnten bereits Ratenmessungen für verschiedene Signale durchgeführt werden. Dabei wurden die Raten von den beiden SiPM-Signalen, dem Triggersignal (Koinzidenz der beiden SiPM-Signale) und dem Anodensignal aufgezeichnet. Dazu werden für jeden Messschritt für eine bestimmte von der Art der Messung abhängige Zeit die Signale gezählt und die Rate ermittelt. Verschiedene Parameter wurden für die Messungen variiert:

1. Beide SiPMs werden mit derselben Spannung (68V-71V) versorgt. Da die SiPMs in Pärchen mit möglichst kleiner Differenz im Arbeitspunkt gewählt sind, wird in dieser Messung der beste Arbeitspunkt bei gleicher Spannung ermittelt (Abb. 7.6). Die Messung legt einen Arbeitspunkt für VDC2 von 69,5 nahe, da hier die SiPM-Raten gegenüber dem Trigger nur wenig größer sind und das Verhältnis von Trigger und zufälliger Koinzidenz sehr gut ist.
2. Beide SiPMs werden mit der aus Messung 1 ermittelten Arbeitsspannung versorgt. Bei unveränderter Spannung werden mehrfach Ratenmessung durchgeführt. Diese liefern einen Eindruck der Zeitstabilität der Raten (Abb. 7.7). Trotz der stark schwankenden SiPM-Raten ist die zeitliche Konstanz der Anoden- und Trigger-Rate sehr gut. Die Schwankungen der SiPM-Raten sind auf die Temperaturunterschiede im Laufe des Tages zurückzuführen.

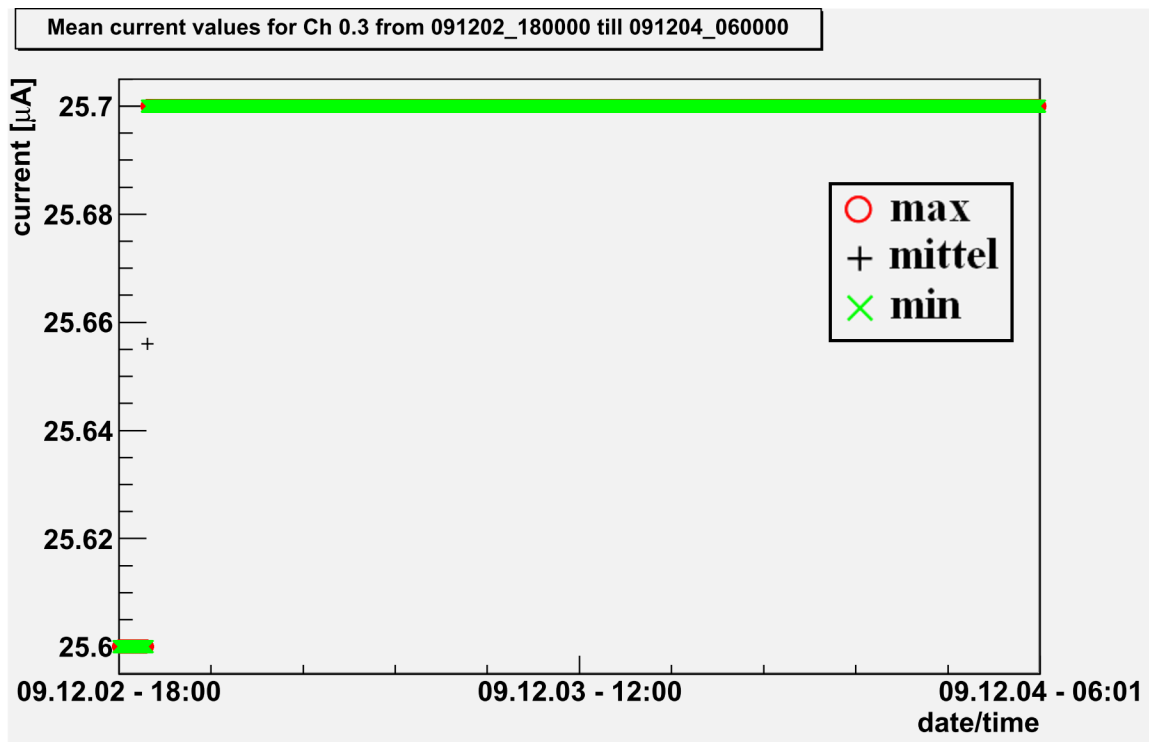


Abbildung 7.2.: Zwei-Tages-Ausschnitt aus dem Graph des Stroms einer Kammer ohne Auffälligkeiten. Die Schwankungen des Stroms sind  $< 0,1 \mu\text{A}$ . (Kathodenstrom VDC4; ca. 20000 Messpunkte, Digitalisierungsschritt  $0,1 \mu\text{A}$ )

3. Beide SiPMs werden mit verschiedenen Spannungen versorgt. SiPM1 erhält die Arbeitsspannung, während die Spannung bei SiPM2 variiert wird. Liefert einen möglicherweise besseren Arbeitspunkt für SiPM2 (Abb. 7.8). Der Graph legt nahe, den Arbeitspunkt von SiPM2 um  $0,1 \text{ V}$  auf  $69,4 \text{ V}$  zu reduzieren.
4. Erfolgt analog zu Messung 3, nur mit vertauschten Rollen von SiPM1 und SiPM2 (Abb. 7.9). Der Graph legt weiterhin einen Arbeitspunkt von  $69,5 \text{ V}$  nahe. Die Messung 1 ist daher bei vorausgewählten SiPM-Pärchen eine gute Methode, um einen ersten Arbeitspunkt zu bestimmen.
5. Beide SiPMs werden mit ihrer Arbeitsspannung aus Messung 3 und 4 versorgt. Zusätzlich wird die Faser der Triggerbox mit einer LED mit eingestellter hoher Rate gepulst. Die Testpulsrate ist ca. 1000 mal größer als die der Quellen. Diese Messung erlaubt eine Aussage über die Effizienz der Triggerbox durch den Vergleich von erzeugten und detektierten Pulsen (Abb. 7.10). Wie dem Graphen zu entnehmen ist, beträgt die Effizienz für die gewählte Testpulsform ( $4,8 \text{ V}, 25 \text{ ns}$ ) 100%.
6. Beide SiPMs werden mit ihrer Arbeitsspannung aus Messung 3 und 4 versorgt. Die Anodenspannung wird durchgeföhren (Abb. 7.11). Die Anodenraten variieren trotz baugleicher Kammern, so dass für jede Kammer ein eigener Arbeitspunkt bestimmt werden muss. Bei

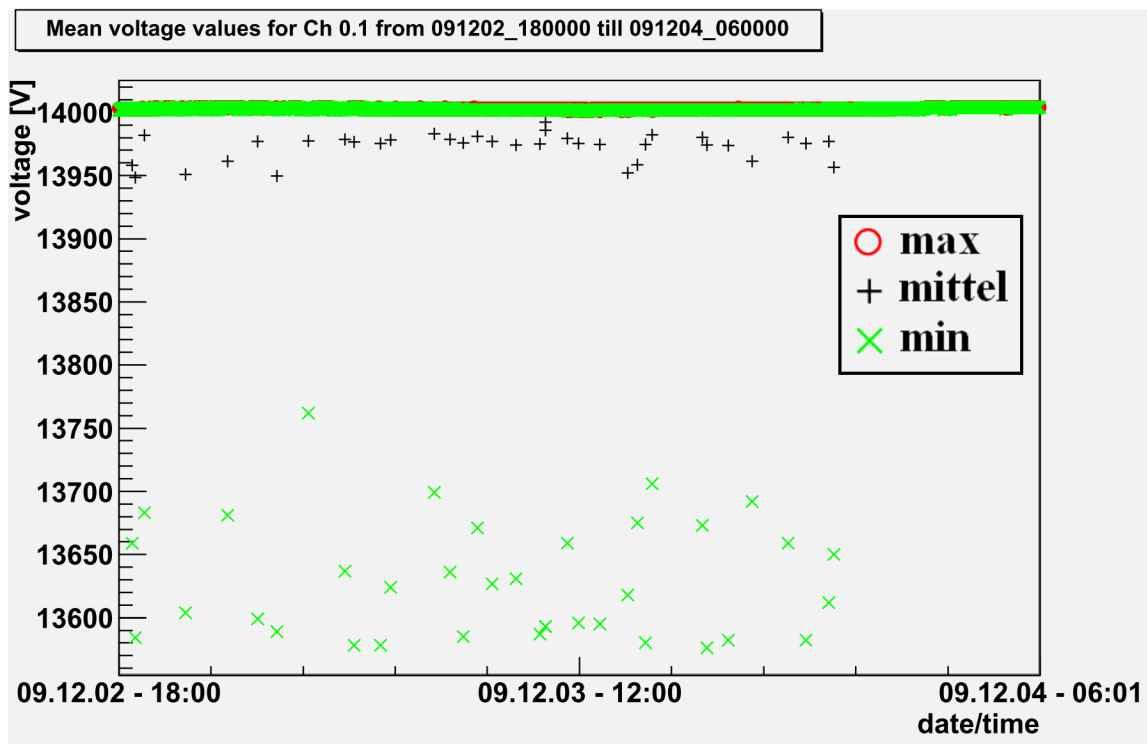


Abbildung 7.3.: Zwei-Tages-Ausschnitt aus dem Graph der Hochspannung einer Kammer mit Störung. Die gelegentlichen kurzzeitigen Schwankungen der Spannung betragen bis zu etwa 400 V. (Kathodenspannung VDC2; ca. 20000 Messpunkte; 35 davon zeigen eine deutlich zu niedrige Spannung, Digitalisierungsschritt 1 V)

VDC2 empfiehlt sich ein Arbeitspunkt zwischen 1850 V und 1950 V. Für VDC5 sollte der Arbeitspunkt zwischen 1800 V und 1880 V gewählt werden.

Für eine leicht höhere Ausbeute an Triggersignalen bei sich fast nicht veränderndem Verhältnis von Trigger zu zufälliger Koinzidenz können die Arbeitspunkte auch um ein paar zehntel Volt angehoben werden. Eine andere Möglichkeit ist es, die Arbeitspunkte solange zu erhöhen, bis eine bestimmte Schwelle im Verhältnis von Trigger zu zufälliger Koinzidenz erreicht wird. Dabei wird aber das Signal mit zusätzlichen Ereignissen verunreinigt, die bei der Analyse wieder herausgefiltert werden müssen.

### 7.3. Temperaturmessungen für die Ventilüberwachung

Zur Optimierung der Ventilüberwachung wurde eine Testmessung gemacht, um den Temperaturverlauf während einer Heizphase zu erfassen.

Aus früheren Tests hatte man eine Heizzeit von 20 s ermittelt, die auch weiterhin als angemessen anzusehen ist (Abb. 7.12). Um die Gefahr einer Überhitzung zu vermeiden, ist im Steuereinschub „VALVE\_READOUT“ neben der Heizzeit von 20 s auch eine Totzeit eingebaut. Aus der Messung mit wiederholten Heizzyklen (Abb. 7.13) erkennt man, dass 60 s als Totzeit ausreichend

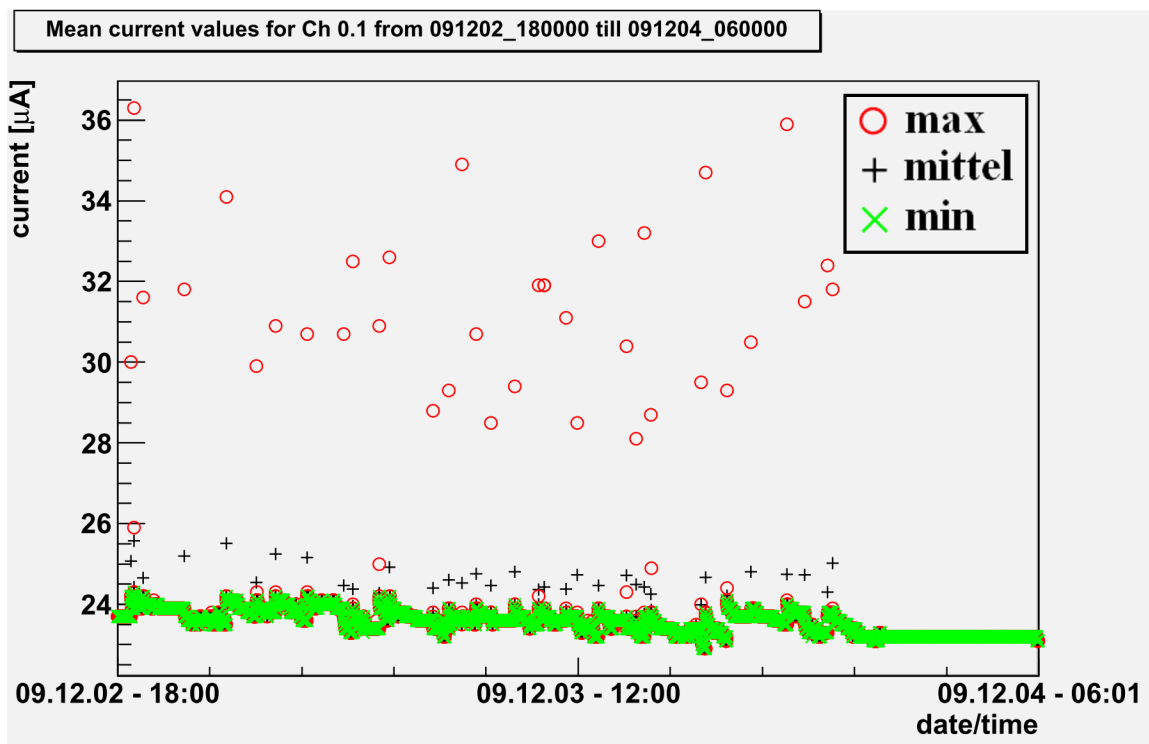


Abbildung 7.4.: Zwei-Tages-Ausschnitt aus dem Graph des Stroms einer Kammer mit Störung. Die Schwankungen des Stroms betragen etwa  $12 \mu\text{A}$ . (Kathodenstrom VDC2; ca. 20000 Messpunkte; 35 davon zeigen einen deutlich zu hohen Strom, Digitalisierungsschritt  $0,1 \mu\text{A}$ )

sind. Die Abbildung verdeutlicht ebenso die Notwendigkeit, für die Erkennung eines Heizpulses Temperaturänderungen anstatt absoluter Temperaturen zu betrachten. Die aktuelle Version des „VALVE\_READOUT“ hat eine Heizdauer von 30 s und 120 s Totzeit.

Die Auslese der Ventilpositionen mittels Temperatursensoren funktioniert demnach zuverlässig. Beim endgültigen Betrieb, wenn Wasserkühlung und alle Geräte in Betrieb sind, sollte diese Messung erneut vorgenommen werden.

## 7.4. Driftgeschwindigkeitsmessungen

Zur Bestimmung der Driftgeschwindigkeit werden die vom TDC erfassten Ereignisse analysiert und ausgewertet. In die gezeigten Histogramme des VDC-Systems sind nur solche Ereignisse eingeflossen, die mindestens die folgenden vier Kanäle enthielten: PM1, PM2, TRIGG, ANODE. Für die Zeitbestimmung wird immer das erste Anodensignal eines Ereignisses genommen, damit wird ein eventueller Nachpuls der Anode in der Auswertung effektiv unterdrückt. Die Betriebsparameter für die Kammern waren  $-14 \text{ kV}$  für die Kathoden,  $1850 \text{ V}$  für die Anoden und die Triggerboxen auf die jeweiligen Arbeitspunkten. Die Schwellenwerte der Diskriminatoren für SiPM- und Anoden-Signale standen auf  $0,5 \text{ V}$ . Das Driftzeitspektrum von VDC4 (Abb. 7.14)

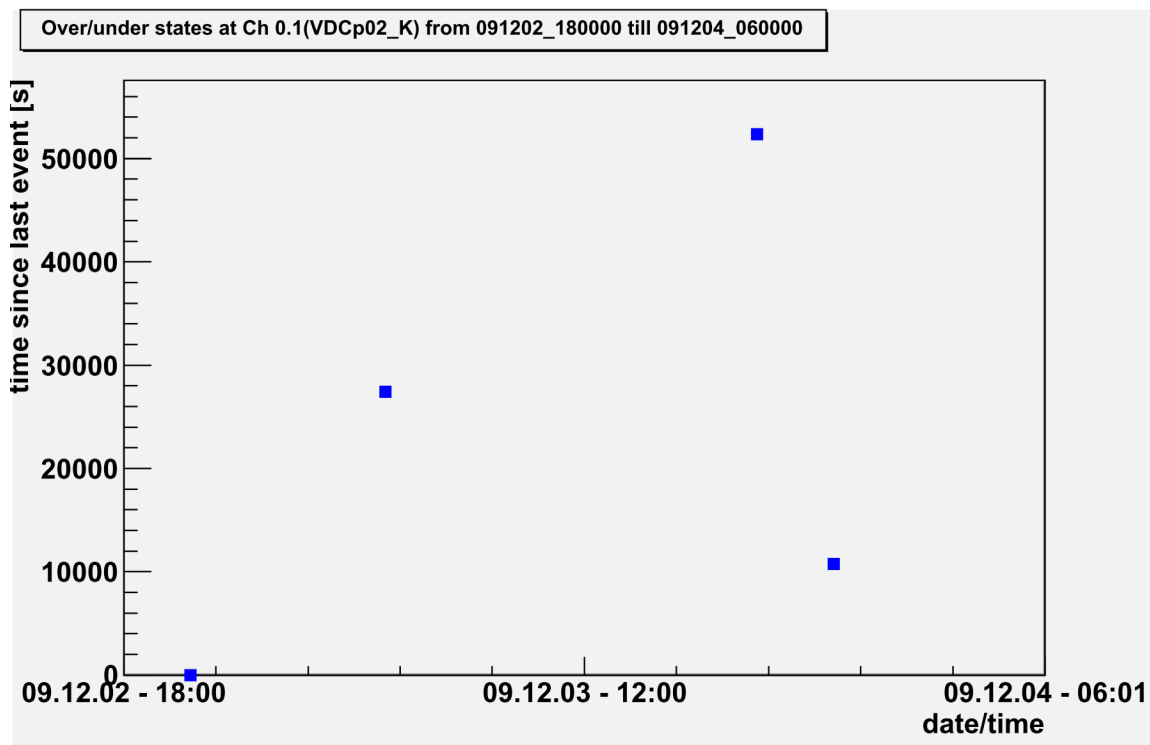


Abbildung 7.5.: Zwei-Tages-Ausschnitt aus dem Graph der Over/Understates einer Kammer mit Störung. Hier sind nur vereinzelte Einträge zu erkennen. (VDC2)

entspricht der Erwartung, die man an diese Messung hat. Es gibt zwei deutliche Spitzen und fast keinen Untergrund. Das Driftzeitspektrum von VDC6 (Abb. 7.15) zeigt ein ganz anderes Bild. Es gibt keine klaren Spitzen und insgesamt einen sehr viel größeren Untergrund. Zum Zeitpunkt dieser ersten Datennahme waren noch drei weitere Kammern in Betrieb. Für diese wurde jedoch kein einziges vollständiges Ereignis im TDC erfasst. Es fehlen alle Anodensignale für diese drei Kammern in den Daten dieser Messung. Bei einer eingehenden Untersuchung aller Kammern wurde festgestellt, dass der Einkoppelkondensator nicht korrekt angelötet war und die daraus resultierende mechanische Belastung durch den Kondensator auf die Lötinsel bei einigen Kammern zum Abriss derselben führte. Diese ungünstige Verbindung wird nun über ein flexibles Kabel hergestellt. Zusätzlich wurde bei zwei Kammern ein Bauteil des Vorverstärkers erneuert. Nach der Reparatur wurden alle fünf vorhandenen Kammern erfolgreich ausgelesen. Auch die Spektren von VDC6 sehen jetzt normal aus. Die ersten fünf simultan aufgenommenen Driftzeitspektren inklusive Driftgeschwindigkeiten werden in Abb. 7.16 gezeigt.

Die rote und die grüne Kurve (in Abb. 7.14 und in Abb. 7.16) sind Gaussianpassungen an die jeweiligen Spitzen. Die Mittelwerte der Gaussianpassungen liefern die beiden Zeiten  $t_1$  und  $t_2$ . Die angegebenen Fehler auf die Zeiten sind die Fehler auf die beiden Mittelwerte der Gaussverteilungen. Durch die Verwendung des Mittelwertes aus einer Verteilung wird eine gegenüber der Einzelmessung deutlich bessere Zeitauflösung erreicht. Der Fehler auf die Driftstrecke ergibt sich aus den Fertigungstoleranzen. Sowohl die Position der Einlassöffnung als auch die der Nut, in



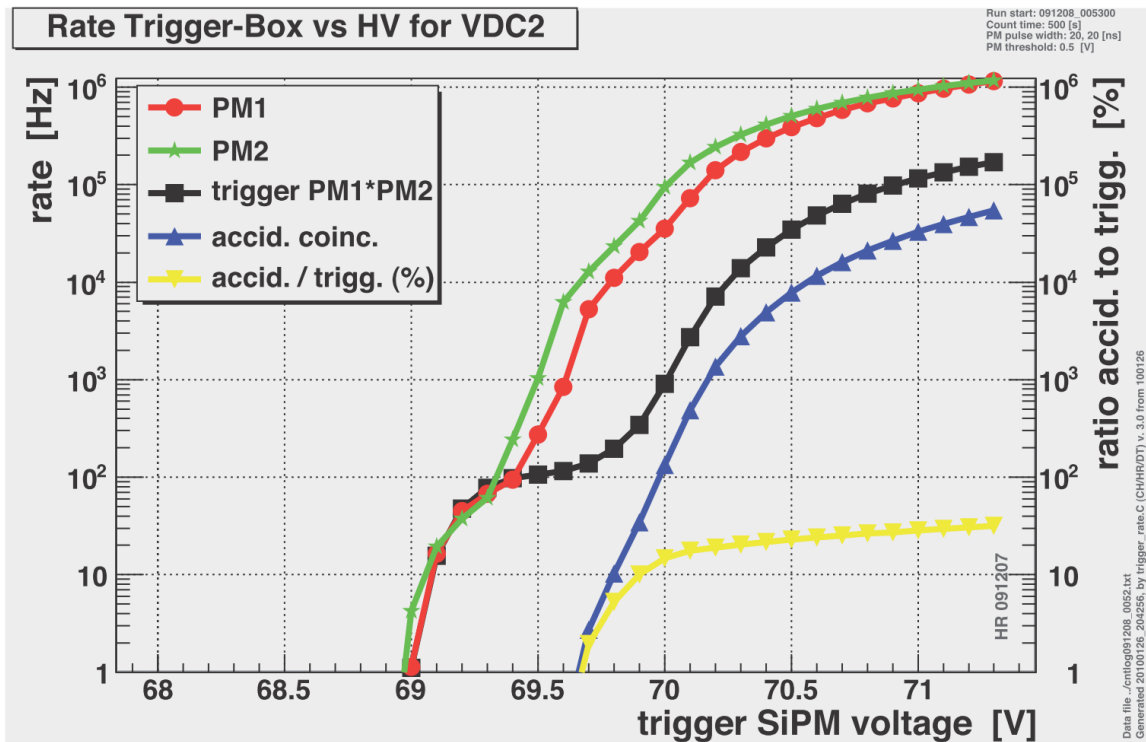


Abbildung 7.6.: Graph[4] der Raten an den SiPMs als Funktion ihrer Arbeitsspannung (Messung 1). Die Messung legt nahe, einen Arbeitspunkt um die 69,5 V zu wählen, da hier die SiPM-Raten gegenüber dem Trigger nur wenig größer sind und das Verhältnis von Trigger zu zufälliger Koinzidenz sehr gut ist.

die die Faser innerhalb der Triggerbox eingelassen ist, lässt sich mit einer Fertigungstoleranz<sup>47</sup> von maximal 20  $\mu\text{m}$  nach oben abschätzen. Die Teile wurden mit deutlich höherer Präzision gefertigt. Der Abstand zwischen beiden Strahlen beträgt nominal 48 mm. Mit diesen Fehlern lassen sich die Fehler der beiden Abstände Quelle-Quelle und Faser-Faser berechnen und damit eine obere Grenze für den Beitrag des Fehlers der Geometrie zum Fehler der gemessenen Driftgeschwindigkeit abschätzen. Die echte Driftstrecke  $s$  liegt mittig zwischen den beiden und hat daher den Mittelwert der beiden als Länge. Den Fehler auf die Driftstrecke erhält man dann durch Fehlerfortpflanzung:

$$\sigma_{QQ} = \sigma_{FF} = \sqrt{2} \cdot 20 \mu\text{m} = 28,28 \mu\text{m} \quad (7.1)$$

$$s = \frac{s_{QQ} + s_{FF}}{2} = 48000 \mu\text{m} \quad (7.2)$$

$$\sigma_s = \sqrt{\frac{1}{2}\sigma_{QQ}^2 + \frac{1}{2}\sigma_{FF}^2} = 28,28 \mu\text{m} \quad (7.3)$$

<sup>47</sup>Fertigungstoleranzen besitzen kein Konfidenzniveau (CL). Sicherheitshalber wird hier die Fertigungstoleranz der CNC-Fräse (10  $\mu\text{m}$ ) verdoppelt.

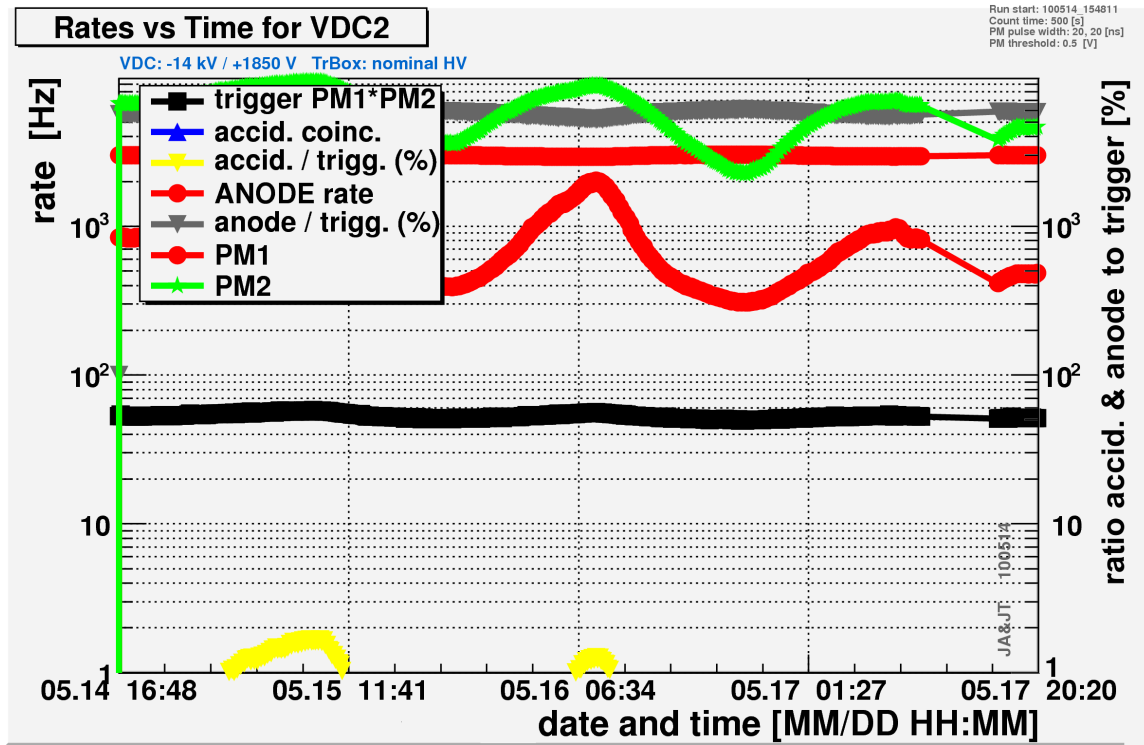


Abbildung 7.7.: Graph[23][24] der Raten an den SiPMs als Funktion der Zeit (Messung 2). Die Messung erfolgte über etwas mehr als 3 Tage. Gut zu erkennen sind die konstanten Trigger- und Anodenraten. Die SiPMs zeigten eine periodische Änderung im 24 h Rhythmus, dies ist auf die Temperaturänderungen von ca. 2 °C im Laufe eines Tages zurückzuführen.

Für die Messung der Abb. 7.14 erhält man damit für  $v_d$  und  $\sigma_{v_d}$ :

$$v_d = \frac{s}{t_2 - t_1} = 54,688 \mu\text{m/ns} \quad (7.4)$$

$$\sigma_{v_d} = \sqrt{\left(\frac{\sigma_s}{t_2 - t_1}\right)^2 + \left(\frac{\sigma_{t_1} \cdot s}{-(t_2 - t_1)^2}\right)^2 + \left(\frac{\sigma_{t_2} \cdot s}{-(t_2 - t_1)^2}\right)^2} = 0,048 \mu\text{m/ns} \quad (7.5)$$

Der relative Fehler durch die Fertigungstoleranzen liegt also deutlich unter 0,06%. Der relative Fehler durch die Zeitmessung liegt für die Messung, die in Abb. 7.14 gezeigt ist, bei 0,065%. Der Fehler aus der Konstruktionsgenauigkeit ist hier nicht der dominierende Fehler, da es sich hierbei um die Höchstgrenze dieses Beitrags handelt. Die Fehler von Geometrie und Zeitmessung wurden hier als statistische Fehler kombiniert, d.h. quadratisch addiert. Für die einzelne VDC ist der Geometriefehler jedoch ein konstanter, systematischer Fehler, der zu dem Fehler aus der Zeitmessung zu addieren ist. Dies führt dann zu einem Gesamtfehler von höchstens  $0.059\% + 0.065\% = 0,124\%$ , entsprechend zu einer Driftgeschwindigkeit von  $54,69 \pm 0.07 \mu\text{m/ns}$ <sup>48</sup>. Vergleicht man die Driftgeschwindigkeiten aus der ersten simultanen Messung für die vier Kammer 1, 2, 4 und 6, die unter gleichen Bedingungen liefen, so liegen drei (1,4,6) von diesen in einem 1- $\sigma$ -Bereich. Kammer 2

<sup>48</sup>Die Abweichung zur Angabe im Graphen ist auf die höhere Genauigkeit während der Auswertung zurückzuführen, da nur die Ausgaben gerundet werden und alle Rechnungen mit voller Genauigkeit ausgeführt werden.

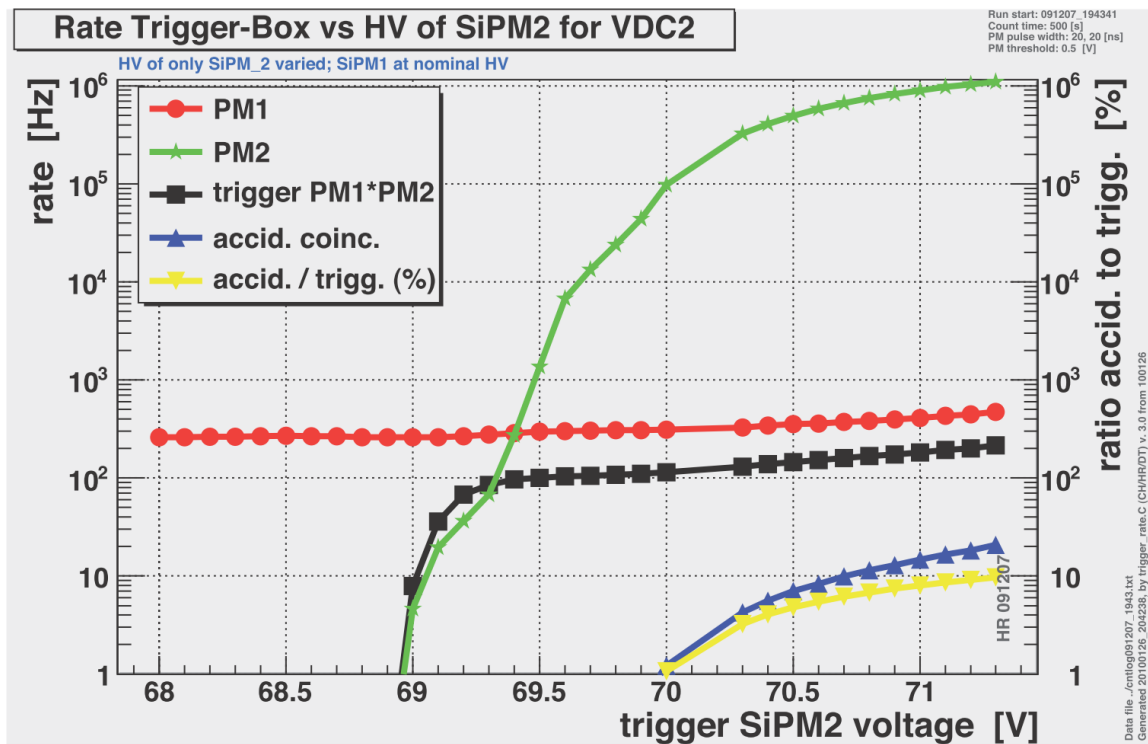


Abbildung 7.8.: Graph[4] der Raten an den SiPMs als Funktion der Arbeitsspannung von SiPM2 (Messung 3). Der Verlauf des Graphen von SiPM2 legt nahe, die Spannung für diesen SiPM auf 69,4 V zu setzen.

liegt in einen 2- $\sigma$ -Berich von Kammer 4. Insgesamt zeigen diese vier Kammern ein homogenes Bild der Driftgeschwindigkeiten, trotz der noch unbekannt individuellen systematischen Fehler der einzelnen Kammern. Eine erste vorsichtige Abschätzung der systematischen Fehler aus dieser Messung deutet an, dass  $\sigma_{sys} \leq 0,1 \mu\text{m}/\text{ns}$  ist.

## 7.5. Abschließende Bemerkungen

Die Messungen zeigen, dass man auf dem richtigen Weg ist und der große Sprung von Messungen mit einer einzelnen Kammer zur simultanen Messung mit sechs Kammern geschafft wurde. Die Regelung bzw. Überwachung der weiteren relevanten Parameter, die bislang nicht permanent per PC überwacht wurden, wie Gasfluss und Gasdruck, Ventilstellungen, Temperaturen und Zählraten von Signalen befinden sich derzeit in Vorbereitung. Man wird also in Kürze in der Lage sein, mit systematischen Messungen des VDC-Systems zu beginnen.

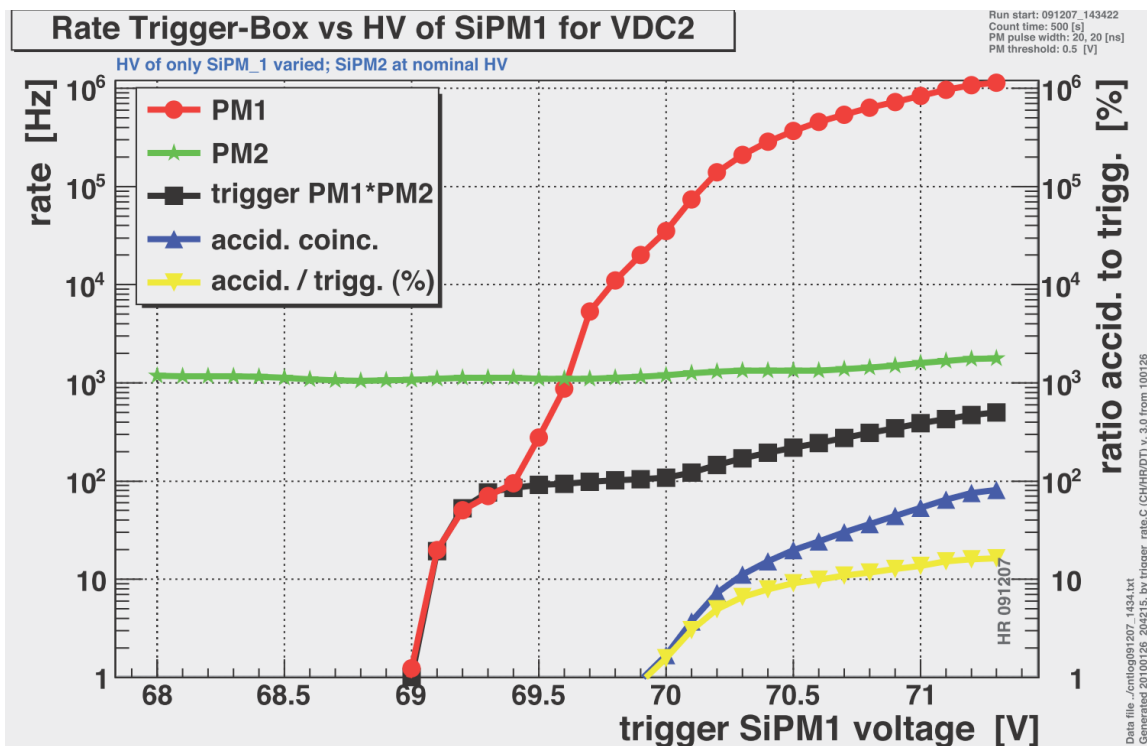


Abbildung 7.9.: Graph[4] der Raten an den SiPMs als Funktion der Arbeitsspannung von SiPM1 (Messung 4). Der Verlauf des Graphen von SiPM1 legt nahe, die Spannung für diesen SiPM auf 69,5 V zu setzen.

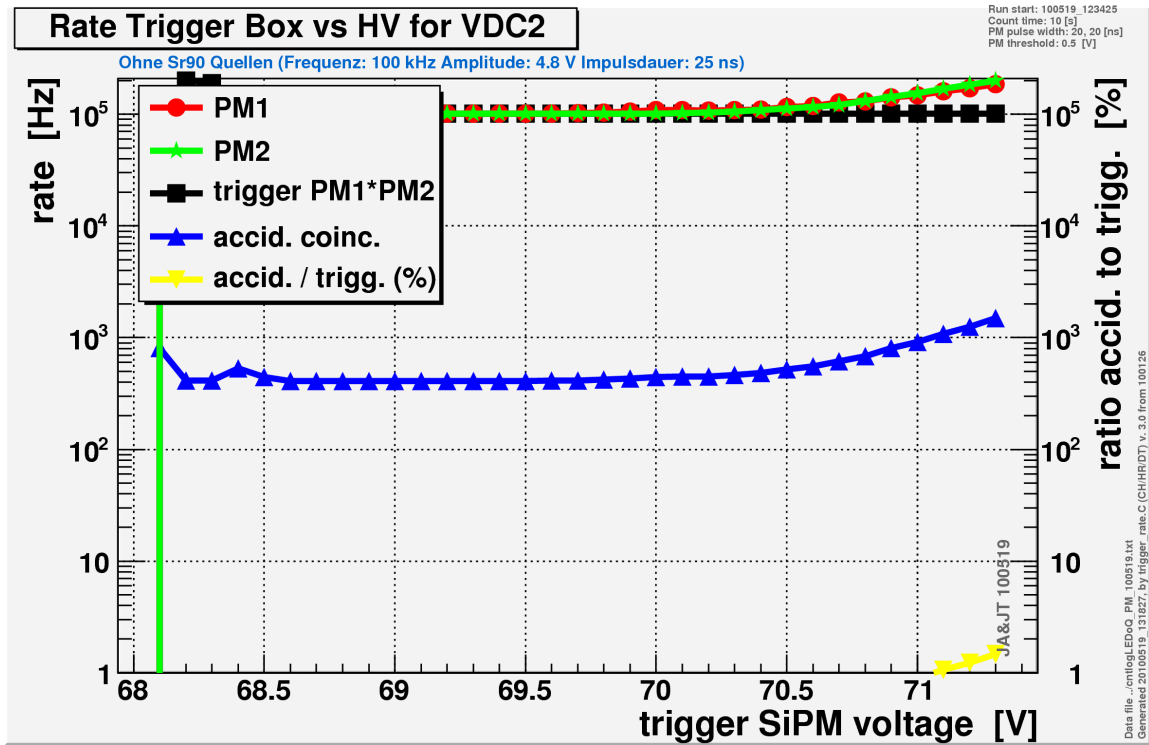


Abbildung 7.10.: Graph[23][24] der Raten an den SiPMs mit optischen Testpulsen als Funktion der Arbeitsspannung (Messung 5). Die Effizienz der SiPMs liegt in dieser Messung bei 100%, zu erkennen an der konstanten Triggerrate, die der eingespeisten Lichttestpulsrate entspricht.

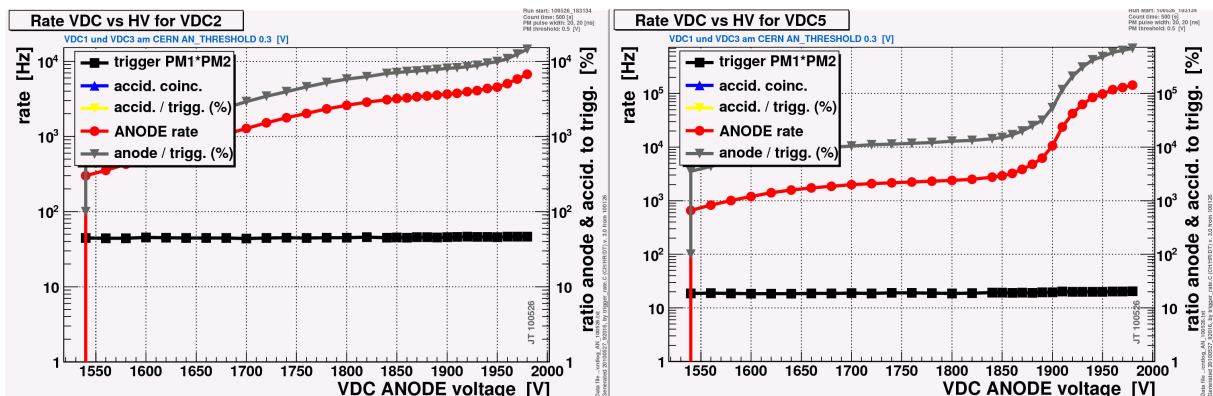


Abbildung 7.11.: Graphen[24] der Raten an der Anode als Funktion der Anodenspannung (Messung 6). Es ist gut zu erkennen, dass die Anodenspannung für VDC2 im Bereich des Plateau von 1850 V bis 1950 V gewählt werden sollte. Bei der Messung von VDC5 lässt sich ein weiteres Plateau oberhalb von 2000 V erahnen. Ob diese Sättigung durch die steigende Pulslänge bedingt ist, konnte noch nicht verifiziert werden.

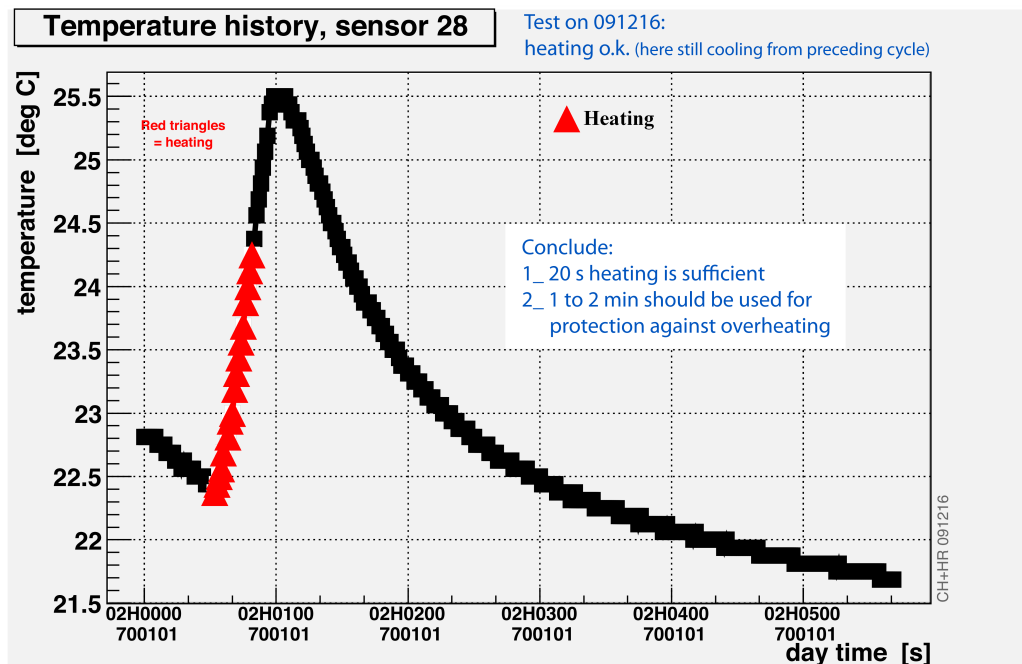


Abbildung 7.12.: Graph der Temperatur, der zur Bestimmung einer sinnvollen Heizzeit herangezogen wurde. Man erkennt die Fortsetzung des Temperaturanstieges nach dem Abschalten der Heizung. Die Zeit, in der geheizt wurde, ist durch die roten Dreiecke markiert.

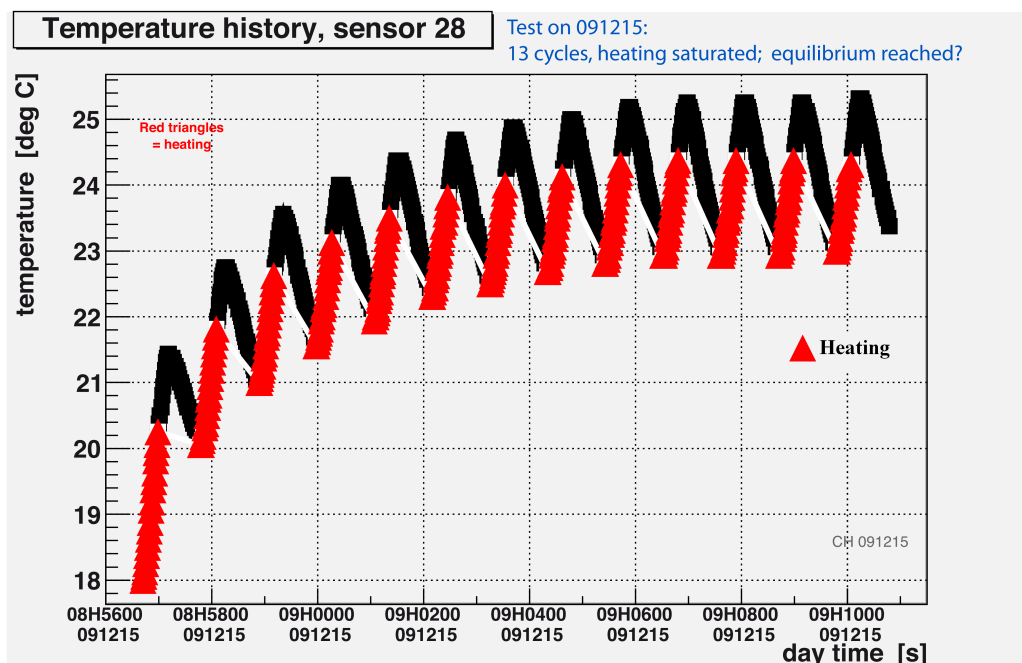


Abbildung 7.13.: Graph der Temperatur, der zur Bestimmung einer sinnvollen Totzeit herangezogen wurde. Ein Zyklus von 20 s Heizen und 60 s Abkühlen (Totzeit) führt zu einer maximalen Temperatur von 25,5 °C. Diese Temperatur sollte von allen verwendeten Materialien auch auf Dauer vertragen werden.

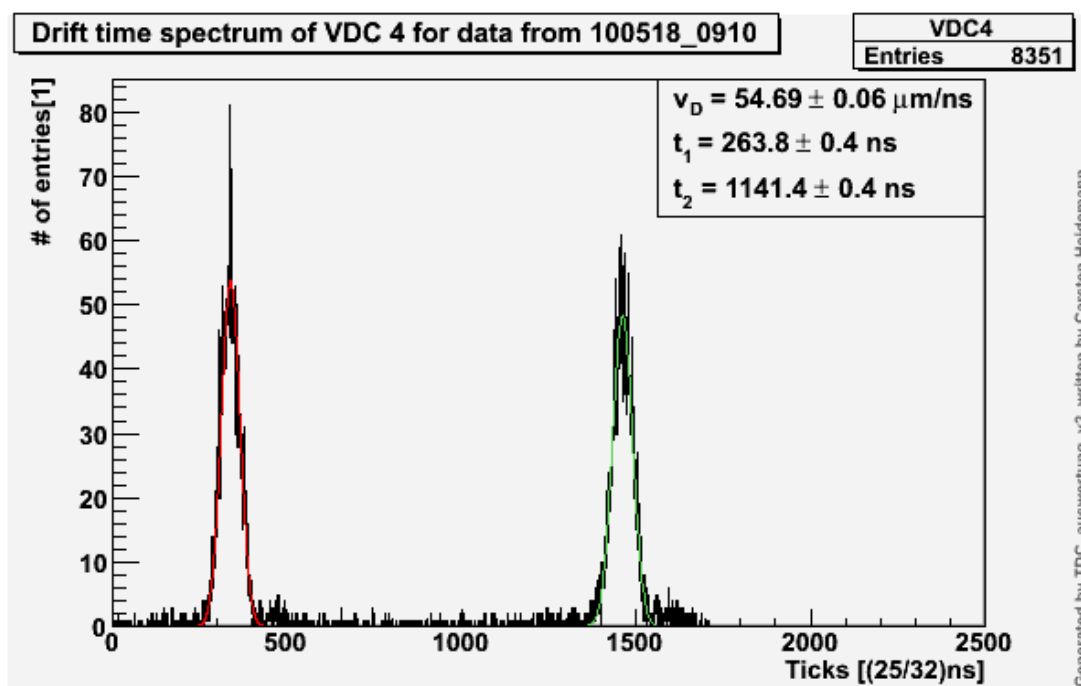


Abbildung 7.14.: Das Driftzeitspektrum von VDC4 zeigt zwei deutliche Peaks und einen sehr niedrigen Untergrund. Die Intervallbreite des Histogramms beträgt 1 Tick. Der Fehler auf die Driftgeschwindigkeit ergibt sich als lineare Addition des für eine Kammer systematischen Ortsfehlers und des statistischen Fehlers aus der Zeitmessung.

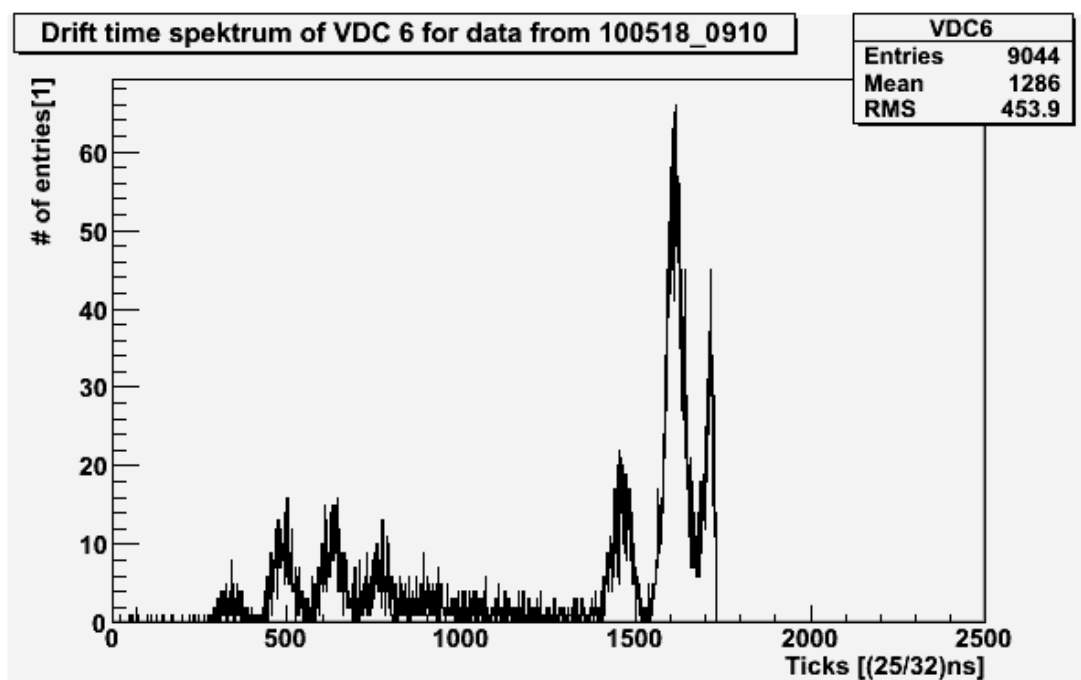


Abbildung 7.15.: Das Driftzeitspektrum von VDC6 zeigt keine eindeutigen Peaks. Es lässt sich nicht klar zwischen Untergrund und Signal unterscheiden. Eine sinnvolle und korrekte Bestimmung der Driftgeschwindigkeit ist mit diesem Graphen nicht möglich.

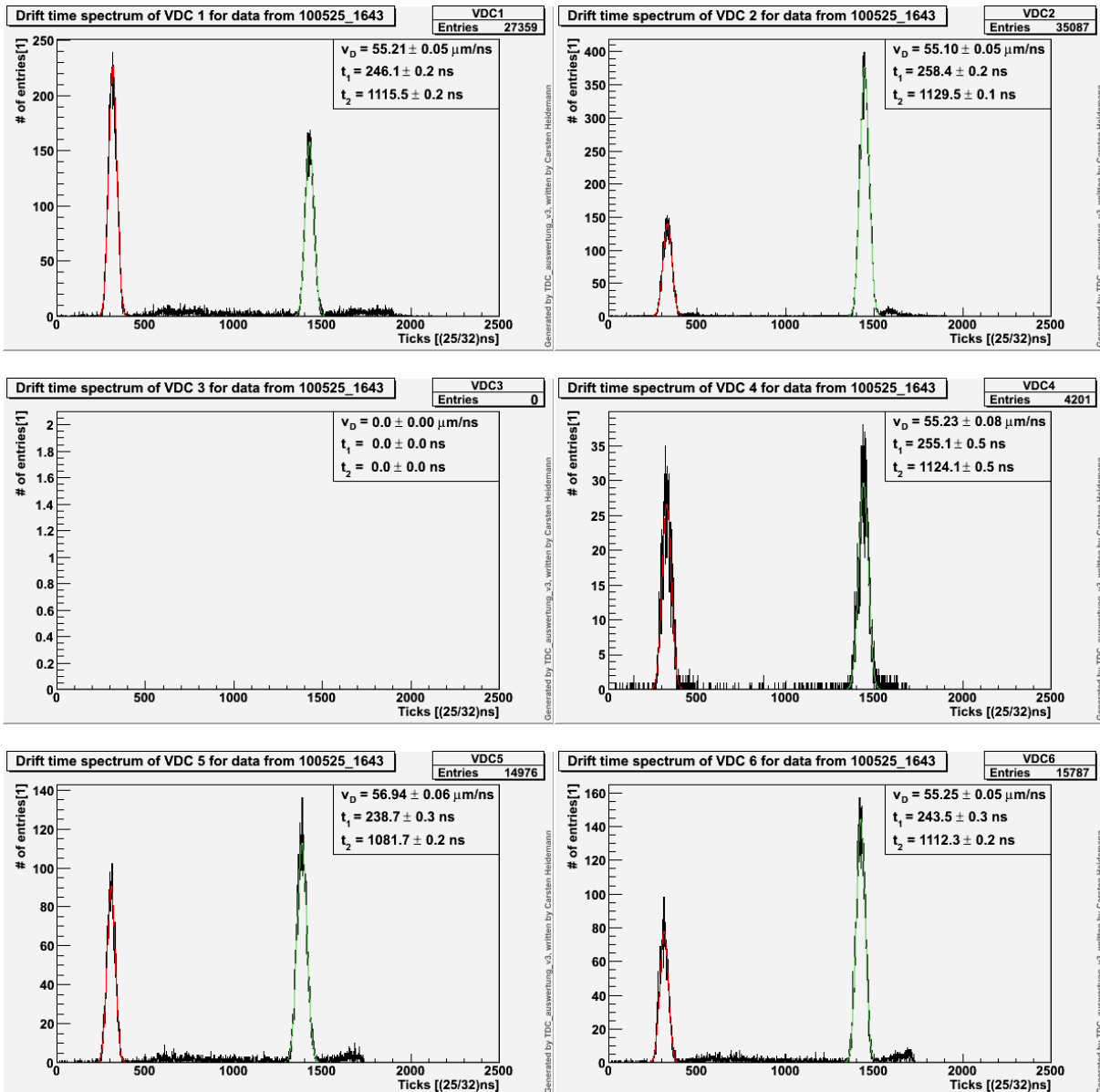


Abbildung 7.16.: Simultane Messung von fünf Driftgeschwindigkeiten. Durch die simultane Messung des selben Gases unter möglichst gleichen Bedingungen kann der systematische Fehler der VDCs ermittelt werden. Die Messungen hier zeigen das selbe Gas unter gleichen Bedingungen. Da VDC3 nicht installiert war, konnten für diese Kammer keine Daten genommen werden.



## 8. Fazit und kritische Bewertung

Ziel dieser Arbeit war es, dass Aachener VDC-System in Betrieb zu nehmen, indem die benötigte Software und Firmware entwickelt wird.

Man ist jetzt in der Lage mit mehreren Kammern gleichzeitig zu messen und viele der relevanten Parameter zu überwachen. Die systematischen Tests können jetzt beginnen, dabei sollen insbesondere die systematischen Fehler der einzelnen Kammern ermittelt werden.

Zu Beginn der vorliegenden Diplomarbeit bestand ihr Hauptziel darin, die Firmware für die im Institut entwickelten eigenen Koinzidenz- und Diskriminatoremodule zu erstellen und zu testen. Jetzt besitzt man zwei Prototypen des COINC3x2-Moduls. An diesen wurden mit den erstellten Programmierungen bereits viele entscheidende Tests erfolgreich durchgeführt. Mit den nächsten Tests und deren endgültiger Programmierung muss gewartet werden, da den vorliegenden Prototypen eine wichtige, später entwickelte Erweiterung noch fehlt.

Die Überwachungs- und Steuerungssoftware für das VDC war zu Beginn nicht existent. Im Rahmen dieser Arbeit wurden erste Versionen zur Überwachung und Steuerung von Hochspannung, TDC, Temperatur und Zählern erstellt. Im Zuge einer Bachelorarbeit werden derzeit grafische Oberflächen, Software für Druckmessung und Druck- und Flussregelung erstellt, sowie die Temperaturüberwachung verfeinert. Anfangs befanden sich nur der Hochspannungsrahmen, die sechs Kammern und der PC im Schrank, im Laufe der Zeit wurden nach und nach alle Komponenten installiert. Als nächstes wird noch der einzelne Prototyp eines Pulsers durch vier verbesserte Pulsler ersetzt.

Insgesamt ist das Projekt während dieser Diplomarbeit durch das Mitwirken aller Beteiligten entscheidend vorangeschritten.

Ich möchte an dieser Stelle noch deutlich darauf hinweisen, dass nur die Programme von mir entwickelt wurden. Die Elektronikentwicklung und die mechanischen Arbeiten wurden von anderen Personen ausgeführt.

## Danksagung

Ich danke Prof. Dr. Thomas Hebbeker für die Möglichkeit, diese Diplomarbeit bei ihm zu erstellen. Bei Dr. Hans Reithler möchte ich mich ganz herzlich für die hervorragende Projekt- und Arbeitsbetreuung bedanken. Mein Dank gilt auch den Mitarbeitern der mechanischen Werkstatt, insbesondere Hans Frohn, Josef Grooten und Barthel Philipps und den Mitarbeitern der elektrischen Werkstatt, insbesondere Rainer Lampe, Franz Adamczyk und Günter Hilgers. Ein Dankschön gehört auch den alten und neuen Kollegen der VDC-Gruppe Aachen, nämlich Jennifer Arps, Lukas Koch, Lars Sonnenschein, Daniel Teyssier und Jenny Tempeler sowie meinen beiden Testlesern Markus Merschmeyer und Arndt Meyer.

## Literaturverzeichnis

- [1] LHC@home, “Overall view of the LHC”, <http://lhcatome.cern.ch/lhc/lhc.shtml>, 1997.  
(Referenziert : 1.1)
- [2] The CMS Collaboration and S. Chatrchyan et al, “The CMS experiment at the CERN LHC”, Journal of Instrumentation, <http://dx.doi.org/10.1088/1748-0221/3/08/S08004>, 2008.  
(Referenziert : 1.2, 2 und 1.3)
- [3] The CMS Collaboration and G.L. Bayatian et al, „CMS The Hadron Calorimeter Technical Design Report“, CERN/LHCC 97-31 CMS TDR2, 1997. (Referenziert : 3)
- [4] H. Reithler, private communication , Physikalisches Institut IIIA der RWTH Aachen, 2010.  
(Referenziert : 1.4, 1.5, 3.1, 3.6, 3.7, 3.10, 5.1, 7.6, 7.8 und 7.9)
- [5] C. Amsler et al. (Particle Data Group), “Review of Particle Physics”, Physics Letter B667, 2008. (Referenziert : 2.1, 2.2 und 2.1.3)
- [6] H. J. Leisi, “Quantenphysik: Eine Einführung anhand elementarer Experimente”, Springer, 2006. (Referenziert : 2.2)
- [7] V. Abazov et al., “Search for Higgs Boson Production in Dilepton and Missing Energy Final States with 5.4 fb<sup>-1</sup> of ppbar Collisions at s=1.96 TeV”, Phys. Rev. Lett. Vol.104 Nr.6, <http://dx.doi.org/10.1103/PhysRevLett.104.061804>, 2010. (Referenziert : 2.2)
- [8] R. Barate et al. (ALEPH, DELPHI, L3, OPAL Collaborations, and The LEP Working Group for Higgs Boson Searches), “Search for the Standard Model Higgs boson at LEP”, Phys. Lett. B 565, 61 [http://dx.doi.org/10.1016/S0370-2693\(03\)00614-2](http://dx.doi.org/10.1016/S0370-2693(03)00614-2), 2003. (Referenziert : 2.2)
- [9] W. Blum, W. Riegler and L. Rolandi, “Particle detection with drift chambers”, Springer, 2008. (Referenziert : 2.3.1 und 2.3.1)
- [10] C. Grupen, “Particle Detektors”, Cambridge University Press, 2008. (Referenziert : 2.3.2)
- [11] R. C. Fernow, “Introduction to Experimental Particle Physics”, Cambridge University Press, 1986. (Referenziert : 2.3.2)
- [12] G. Altenhöfer, “Development of a Drift Chamber for Drift Velocity Monitoring in the CMS Barrel Muon System”, Diplomarbeit, Physikalisches Institut IIIA der RWTH Aachen, 2006.  
(Referenziert : 3 und 3.2.1.1)

- [13] A. Böhm et al, “Experience with the L3 vertex drift chamber at LEP”, Nucl. Instrum. Methods Phys. Res. A, 2003. (Referenziert : 3)
- [14] L. Baksay et al, “Apparatus for precise drift time change measurements by means of photoelectrons emitted from the cathode”, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, <http://www.sciencedirect.com/science/article/B6TJM-46YCY7Y-13/2/eb8ec798721048061713ff3c6df0c5fc>, 1991. (Referenziert : 3)
- [15] M. Sassowsky, “Überwachung von Driftkammergasen mit einer Testkammer”, Diplomarbeit, III. Physikalisches Institut A der RWTH Aachen, 1990. (Referenziert : 14)
- [16] A. Tavenrath, “Bau einer Driftkammer zur Überwachung der Gasqualität in der L3-Vertexkammer”, Diplomarbeit, III. Physikalisches Institut A der RWTH Aachen, 1989. (Referenziert : 14)
- [17] J. Frangenheim, “Measurements of the drift velocity using a small gas chamber for monitoring of the CMS muon system”, Diplomarbeit, III. Physikalisches Institut A der RWTH Aachen, 2007. (Referenziert : 3.2.1.2)
- [18] M. Sowa, “Aufbau, Kalibration und Anwendung einer Messapparatur zur Überwachung des Gasdrucks in den CMS-Myonkammern”, Diplomarbeit, III. Physikalisches Institut A der RWTH Aachen, 2003. (Referenziert : 3.2.4.2)
- [19] R. Brun and F. Rademakers, “ROOT - An Object Oriented Data Analysis Framework,” Proceedings AIHENP’96 Workshop, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. <http://root.cern.ch/>, 1996. (Referenziert : 4.1)
- [20] L. Koch, “Überwachungs- und Steuerungssoftware der Betriebsparameter des VDC-Systems bei CMS”, Bachelorarbeit, III. Physikalisches Institut A der RWTH Aachen, 2010. (Referenziert : 4.3)
- [21] VITA, “ANSI/VITA 1.0-1994 (R2002) VME64”, <http://www.vita.com/specifications.html>, 2002. (Referenziert : 5.4.2.3)
- [22] M. Dworkin, “COM Terminal”, <http://cc5x.de/COM-Terminal/>, 2004. (Referenziert : 6.5)
- [23] J. Arps, “Measurements of the Drift Velocity with Drift Velocity Chambers”, Bachelorarbeit, III. Physikalisches Institut A der RWTH Aachen, 2010. (Referenziert : 7.7 und 7.10)
- [24] J. Tempeler, “Measurements of the Trigger Rate of Drift Velocity Chambers”, Bachelorarbeit, III. Physikalisches Institut A der RWTH Aachen, 2010. (Referenziert : 7.7, 7.10 und 7.11)

## Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Diplomarbeit mit dem Thema

*Direkte Messung der Driftgeschwindigkeit im Gas der CMS-Muonenkammern mit VDC-Kammern*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, den 5. Juli 2010

---

CARSTEN HEIDEMANN

## A. VHDL-Quelltexte

Die folgenden VHDL-Programme wurden mit der Entwicklungsumgebung „ISE Webpack“ von Xilinx erstellt. Die Bezeichnungen am Anfang eines jeden Programms wurden automatisch von der Entwicklungsumgebung erstellt. Der Begriff „Behavioral“, zu deutsch „Verhaltenweise“, klingt zunächst ein wenig schwammig. Jedoch trifft dieser die Bedeutung einer HDL (Hardwarebeschreibungssprache), in der das gewünschte Verhalten (Funktionsweise) einer digitalen Schaltung beschrieben wird. Es handelt sich dabei um den funktionalen Quelltext des VHDL-Programms.

Listing A.1: Vollständiger VHDL-Quelltext des Zählersubmoduls. Eine ausführliche Beschreibung der Funktionsweise dieses Programmcodes ist in Kapitel 5.4.2.1 zu lesen.

```

1 -----
2 -- Company: RWTH–Aachen , Physikalisches Institut 3a
3 -- Engineer: Carsten Heidemann
4 --
5 -- Design Name: Asynchroner Zähler mit Reset und Lesekopie
6 -- Module Name: async_cnt_reset_read_copy – Behavioral
7 -- Project Name:
8 -- Target Devices: Spartan 3E 500
9 -- Tool versions: ISE 10.1.03
10 -- Description: Zähler mit einstellbarer Breite , default 64bit, keine Totzeit durch Auslese
11 --           Zählt unabhängig vom Systemtakt die Pulse am input in 2 Registern.
12 --           Wenn read_request high, wird nur in cnt weitergezählt, cnt_copy nicht,
13 --           damit Ausgabe unverändert, copied wird auf 0 gesetzt.
14 --           Wenn read_request low und reset = 0 und copied = 0 , dann wird cnt_copy = cnt
15 --           gesetzt um eine aktuelle Kopie zu haben unabhängig davon von ob ein
16 --           weiteres input signal kam oder nicht, danach copied = 1, ebenso wie im normalen zählen.
17 -- Dependencies:
18 --
19 -- Revision:
20 -- Revision 1.00 – Getestet auf dem Testboard mit Tastern, Grundlegende Logik funktioniert,
21 --           Highspeed noch nicht getestet
22 -- Revision 0.01 – File created
23 -- Additional Comments: Finally a working solution
24 -----
25 library IEEE;
26 USE IEEE.STD_LOGIC_1164.ALL;
27 USE IEEE.NUMERIC_STD.all;

```

```
28
29 ----- Uncomment the following library declaration if instantiating
30 ----- any Xilinx primitives in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity async_cnt_reset_read_copy is
35   Generic (
36     cnt_breite : integer := 64
37   );
38   Port ( input : in  STD_LOGIC;
39         reset  : in  STD_LOGIC;
40         read_request : in  STD_LOGIC;
41         cnt_enable : in  STD_LOGIC;
42         cnt_out  : out unsigned (cnt_breite-1 downto 0)
43   );
44 end async_cnt_reset_read_copy;
45
46 architecture Behavioral of async_cnt_reset_read_copy is
47   signal cnt, cnt_copy : unsigned(cnt_breite-1 downto 0);
48   signal copied: STD_LOGIC:= '0';
49 begin
50   process (input, reset, read_request, cnt, copied)
51   begin
52     if reset='1' then
53       cnt <= (others => '0');
54       cnt_copy <=(others => '0');
55     elsif input'event and input = '1' then
56       if cnt_enable='1' then
57         cnt <= cnt + 1;
58         if read_request = '1' then
59           cnt_copy <= cnt_copy;
60           copied <= '0';
61         else
62           cnt_copy <= cnt + 1;
63           copied <= '1';
64         end if;
65       end if; -- cnt_ena
66     end if; -- reset / input
67     -- Kopie anfertigen auch wenn kein weiterer input puls gekommen ist
68     if read_request = '0' and reset = '0' and copied = '0' then
69       cnt_copy <= cnt;
70       copied <= '1';
71     end if ;
72   end process;
73   cnt_out <= cnt_copy;
```

74 `end Behavioral;`

Listing A.2: Vollständiger VHDL-Quelltext des Entprellersubmoduls. Dieses Modul leitet einen Zustand eines Tasters oder Schalters erst dann weiter, wenn sich der Zustand über einen gewissen Zeitraum nicht mehr geändert hat.

```

1  -----
2  -- Company: RWTH–Aachen , Physikalisches Institut 3a
3  -- Engineer: Carsten Heidemann
4  --
5  -- Design Name:
6  -- Module Name: entpreller – Behavioral
7  -- Project Name:
8  -- Target Devices:
9  -- Tool versions:
10 -- Description:
11 --
12 -- Dependencies:
13 --
14 -- Revision:
15 -- Revision 1.0 – Tested and working
16 -- Revision 0.01 – File Created
17 -- Additional Comments:
18 --
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ----- Uncomment the following library declaration if instantiating
26 ----- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity entpreller is
31     Port ( clock : in  STD_LOGIC;
32           taster_in : in  STD_LOGIC;
33           status_out : out STD_LOGIC);
34 end entpreller ;
35
36 architecture Behavioral of entpreller is
37 signal taster_old , taster_sampled : std_logic;

```



```

38 signal cnt: std_logic_vector(8 downto 0);
39 constant max_cnt : std_logic_vector := "11111111"; -- 511
40 begin
41 Process(clock)
42 begin
43     if clock ='1' and clock'event then
44         taster_sampled <= taster_in ;
45         if taster_sampled /= taster_old then
46             cnt <= cnt -1;
47         else
48             cnt <= max_cnt;
49         end if ;
50         if cnt = 0 then
51             taster_old <= taster_sampled;
52         end if;
53     end if;
54 End Process ;
55 status_out <= taster_old;
56 end Behavioral;

```

Listing A.3: Vollständiger VHDL-Quelltext des RS232-Interface Submoduls. Dieses Modul stellt die Verbindung zwischen dem Hauptmodul und den beiden anderen Modulen der RS232-Schnittstelle her.

```

1 -----
2 -- Company: RWTH Aachen – III. Physikalisches Institut A
3 -- Engineer: Carsten Heidemann
4 --
5 -- Design Name: Main RS232 Modul
6 -- Module Name: rs232_interface – Behavioral
7 -- Project Name: Logic Modules for VDC
8 -- Target Devices:
9 -- Tool versions:
10 -- Description: interface for the RS232 communication with the three models
11 --
12 -- Dependencies: rs232_receive.vhd , rs232_send.vhd
13 --
14 -- Revision:
15 -- Revision 0.5 – has been tested on test board
16 -- Additional Comments:
17 --
18 -----
19 library IEEE;

```

```

20 USE IEEE.STD_LOGIC_1164.ALL;
21 USE IEEE.NUMERIC_STD.all;
22
23 ----- Uncomment the following library declaration if instantiating
24 ----- any Xilinx primitives in this code.
25 --library UNISIM;
26 --use UNISIM.VComponents.all;
27
28
29 -- Main RS232 Module, transmits / receive the 64bit register
30 -- write
31 -- adr=value+ NewLine , to set reg(adr)=value
32 -- read
33 -- adr + Newline , to read reg(adr)
34 entity reg_trans is
35     generic (sys speed, baudrate : integer);
36     port(
37         reset           : in std_logic;
38         clk             : in std_logic;
39         rx              : in std_logic;
40         tx              : out std_logic;
41         lese_req        : out std_logic;
42         schreib_req     : out std_logic;
43         reg_adr         : out unsigned(7 downto 0);
44         neu_reg_wert    : out unsigned(63 downto 0);
45         aktuell_reg_wert : in unsigned(63 downto 0)
46     );
47 end entity reg_trans;
48
49 architecture Behavioral of reg_trans is
50     constant maxcnt: natural := sys speed / baudrate;
51
52     subtype byte is unsigned(7 downto 0);
53     subtype halbbyte is unsigned(3 downto 0);
54
55     type statusType is (
56         lese_reg_adr_high,
57         lese_reg_adr_low,
58         lese_wert,
59         lese_adr_warten,
60         lese_adr,
61         write_halbbyte,
62         warte_auf_halbbyte,
63         write_lf,
64         warte_auf_neue_zeile);
65

```

```

66  signal status                : statusType := lese_reg_adr_high;
67  signal halbbitzaehler       : natural range 0 to 15;
68  signal adr                   : byte := (others => '0');
69  signal wert                   : unsigned(63 downto 0) := (others => '0');
70
71  signal rs232_empf_byte       : byte := (others => '0');
72  signal rs232_neues_byte_empf : std_logic := '0';
73  signal rs232daten           : byte := (others => '0');
74  signal rs232daten_send_req  : std_logic := '0';
75  signal rs232daten_gesendet  : std_logic := '0';
76
77  -- convert an ASCII hex character to a byte
78  function hexToUnsigned(ascii: byte) return halbbyte is
79  begin
80      if ascii >= x"30" and ascii <= x"39" then
81          return ascii(3 downto 0);
82      elsif ascii >= x"61" and ascii <= x"66" then
83          return ascii(3 downto 0) + x"9";
84      else
85          return x"0";
86      end if;
87  end;
88
89  -- convert a byte to an ASCII hex character
90  function unsignedToHex(data: halbbyte) return byte is
91  begin
92      if data < x"a" then
93          return x"30" + data;
94      else
95          return x"57" + data;
96      end if;
97  end;
98
99  begin
100
101  rs232_receive_instance: entity rs232_receive
102  generic map(
103      sysspeed    => sysspeed,
104      baudrate    => baudrate)
105  port map(
106      reset       => reset,
107      clk         => clk,
108      data_rec    => rs232_empf_byte,
109      data_ready  => rs232_neues_byte_empf,
110      rx         => rx
111  );

```

```

112
113 rs232_send_instance: entity rs232_send
114 generic map(
115     syssspeed    => syssspeed,
116     baudrate     => baudrate)
117 port map(
118     reset        => reset,
119     clk          => clk,
120     data         => rs232daten,
121     tx           => tx,
122     datasendreq  => rs232daten_send_req,
123     datasent     => rs232daten_gesendet
124 );
125
126 process(clk, reset)
127 begin
128     if rising_edge(clk) then
129         rs232daten_send_req <= '0';
130         lese_req           <= '0';
131         schreib_req        <= '0';
132         if reset = '1' then
133             status <= lese_reg_adr_high;
134         else
135             case status is
136                 when lese_reg_adr_high =>
137                     if rs232_neues_byte_empf = '1' then
138                         if rs232_empf_byte > 32 then
139                             adr <= x"0" & hexToUnsigned(rs232_empf_byte);
140                             status <= lese_reg_adr_low;
141                         end if;
142                     end if;
143                 when lese_reg_adr_low =>
144                     if rs232_neues_byte_empf = '1' then
145                         if rs232_empf_byte = x"3d" then
146                             -- "=" received, try to receive new register wert
147                             wert <= (others => '0');
148                             status <= lese_wert;
149                         else
150                             if rs232_empf_byte < 32 then
151                                 -- newline or return received, read adr
152                                 reg_adr <= adr;
153                                 lese_req <= '1';
154                                 status <= lese_adr_warten;
155                             else
156                                 -- next halbyte of adr
157                                 adr <= adr(3 downto 0) & hexToUnsigned(rs232_empf_byte);

```

```
158         end if;
159     end if;
160 end if;
161 when lese_wert =>
162     if rs232_neues_byte_empf = '1' then
163         if rs232_empf_byte < 32 then
164             -- newline or return received, write new wert
165             reg_adr <= adr;
166             neu_reg_wert <= wert;
167             schreib_req <= '1';
168             status <= lese_reg_adr_high;
169         else
170             -- next halbbyte of wert
171             wert <= wert(59 downto 0) & hexToUnsigned(rs232_empf_byte);
172         end if;
173     end if;
174 when lese_adr_warten =>
175     -- wait one more tick
176     status <= lese_adr;
177 when lese_adr =>
178     -- wert available, read it
179     wert <= aktuell_reg_wert;
180     halbbitzaehler <= 15;
181     status <= write_halfbyte;
182 when write_halfbyte =>
183     -- send to RS232 port
184     rs232daten <= unsignedToHex(wert(63 downto 60));
185     rs232daten_send_req <= '1';
186     status <= warte_auf_halfbyte;
187 when warte_auf_halfbyte =>
188     -- wait until character sent
189     if rs232daten_gesendet = '1' then
190         if halbbitzaehler = 0 then
191             rs232daten <= x"0a"; -- LF : 0a
192             rs232daten_send_req <= '1';
193             status <= write_lf; -- warte_auf_neue_zeile;
194         else
195             halbbitzaehler <= halbbitzaehler - 1;
196             wert <= shift_left(wert, 4);
197             status <= write_halfbyte;
198         end if;
199     end if;
200 when write_lf =>
201     if rs232daten_gesendet = '1' then
202         rs232daten <= x"0d"; -- CR : 0d
203         rs232daten_send_req <= '1';
```

```

204         status <= warte_auf_neue_zeile;
205     end if;
206         when warte_auf_neue_zeile =>
207             if rs232daten_gesendet = '1' then
208                 status <= lese_reg_adr_high;
209             end if;
210         end case;
211     end if;
212 end if;
213 end process;
214
215 end architecture Behavioral;

```

Listing A.4: Vollständiger VHDL-Quelltext des RS232-Receive-Subsubmoduls. Dieses Modul empfängt die seriellen Bitsequenzen und speichert sie in einem parallel lesbaren Register.

```

1  -----
2  -- Company: RWTH Aachen – III. Physikalisches Institut A
3  -- Engineer: Carsten Heidemann
4  --
5  -- Design Name: RS232 Receiver
6  -- Module Name: rs232_receive – Behavioral
7  -- Project Name: Logic Modules for VDC
8  -- Target Devices:
9  -- Tool versions:
10 -- Description:  receive submodul for rs232_interface.vhd
11 --
12 -- Dependencies: none
13 --
14 -- Revision:
15 -- Revision 0.2 – beta, has to be tested
16 -- Additional Comments:
17 --
18  -----
19 library IEEE;
20 USE IEEE.STD_LOGIC_1164.ALL;
21 USE IEEE.NUMERIC_STD.all;
22 entity rs232_receive is
23     generic (sys speed,
24             baudrate: integer);
25     port (
26         reset           : in STD_LOGIC;

```

```
27     clk           : in STD_LOGIC;
28     data_rec      : out unsigned(7 downto 0);
29     data_ready    : out std_logic;
30     rx            : in std_logic
31 );
32 end entity rs232_receive;
33
34 architecture Behavioral of rs232_receive is
35     constant maxcnt: natural := syssspeed / baudrate;
36
37     type status_type is (
38         warte_auf_RX_start,
39         warte_auf_halbes_Bit,
40         empfangen_Bits,
41         warte_auf_Stop_Bit);
42
43     signal status      : status_type := warte_auf_RX_start;
44     signal baudratecnt : natural range 0 to maxcnt := 0;
45     signal bitcnt      : natural range 0 to 7 := 0;
46     signal reg_shift   : unsigned(7 downto 0) := (others => '0');
47     signal rx_latch    : std_logic := '1';
48
49 begin
50
51     process(clk, reset)
52     begin
53         if rising_edge(clk) then
54             data_ready <= '0';
55             if reset = '1' then
56                 status <= warte_auf_RX_start;
57                 data_rec <= (others => '0');
58                 data_ready <= '0';
59                 rx_latch <= '1';
60             else
61                 rx_latch <= rx;
62                 case status is
63                     when warte_auf_RX_start =>
64                         if rx_latch = '0' then
65                             status <= warte_auf_halbes_Bit;
66                             baudratecnt <= maxcnt / 2 - 1;
67                         end if;
68                     when warte_auf_halbes_Bit =>
69                         if baudratecnt = 0 then
70                             status <= empfangen_Bits;
71                             bitcnt <= 7;
72                             baudratecnt <= maxcnt - 1;
```

```

73         else
74             baudratecnt <= baudratecnt - 1;
75         end if;
76     when empfangen_Bits =>
77         if baudratecnt = 0 then
78             reg_shift <= rx_latch & reg_shift(7 downto 1);
79             if bitcnt = 0 then
80                 status <= warte_auf_Stop_Bit;
81             else
82                 bitcnt <= bitcnt - 1;
83             end if;
84             baudratecnt <= maxcnt - 1;
85         else
86             baudratecnt <= baudratecnt - 1;
87         end if;
88     when warte_auf_Stop_Bit =>
89         if baudratecnt = 0 then
90             status <= warte_auf_RX_start;
91             if rx_latch = '1' then
92                 data_rec <= reg_shift;
93                 data_ready <= '1';
94             end if;
95         else
96             baudratecnt <= baudratecnt - 1;
97         end if;
98     end case;
99     end if;
100 end if;
101 end process;
102
103 end architecture Behavioral;

```

Listing A.5: Vollständiger VHDL-Quelltext des RS232-Send-Subsubmoduls. Dieses Modul schreibt die Daten aus einem Register seriell auf die RS232-Schnittstelle.

```

1 -----
2 -- Company: RWTH Aachen – III. Physikalisches Institut A
3 -- Engineer: Carsten Heidemann
4 --
5 -- Design Name: RS232 Sender
6 -- Module Name: rs232_send – Behavioral
7 -- Project Name: Logic Modules for VDC
8 -- Target Devices:

```



```
9  --- Tool versions:
10 --- Description:  send submodul for rs232_interface.vhd
11 ---
12 --- Dependencies: none
13 ---
14 --- Revision:
15 --- Revision 0.2 – beta, has to be tested
16 --- Additional Comments:
17 ---
18 -----
19 library IEEE;
20 USE IEEE.STD_LOGIC_1164.ALL;
21 USE IEEE.NUMERIC_STD.all;
22 entity rs232_send is
23
24
25     generic (
26         sys speed,
27         baudrate : integer
28     );
29
30     Port (
31         clk : in  STD_LOGIC;
32         reset : in  STD_LOGIC;
33         data : in  unsigned (7 downto 0);
34         tx : out STD_LOGIC;
35         datasendreq : in  STD_LOGIC;
36         datasent : out STD_LOGIC);
37 end rs232_send;
38
39 architecture Behavioral of rs232_send is
40     constant maxCnt: natural := sys speed / baudrate;
41
42     type status_Type is (
43         warte_auf_strobe,
44         send_bits);
45
46     signal status: status_Type := warte_auf_strobe;
47     signal baudratecnt : natural range 0 to maxcnt := 0;
48     signal bitcnt      : natural range 0 to 10 := 0;
49     signal reg_shift   : unsigned(9 downto 0) := (others => '0');
50
51 begin
52
53     process(clk)
54     begin
55         if rising_edge(clk) then
56             datasent <= '0';
```

```

55     if reset = '1' then
56         tx <= '1';
57         status <= warte_auf_strobe;
58     else
59         case status is
60             -- Warten bis Master gültige Daten hat
61             when warte_auf_strobe =>
62                 if datasendreq = '1' then
63                     -- 1=Stop Bit, Daten , 0= Start Bit ; LSB wird zuerst gesendet
64                     reg_shift <= '1' & data & '0';
65                     bitcnt <= 10;
66                     baudratecnt <= 0;
67                     status <= send_bits;
68                 else
69                     tx <= '1';
70                 end if;
71             when send_bits =>
72                 if baudratecnt = 0 then
73                     baudratecnt <= maxcnt - 1;
74                     tx <= reg_shift(0);
75                     reg_shift <= shift_right(reg_shift, 1);
76                     if bitcnt = 0 then
77                         status <= warte_auf_strobe;
78                         dataSent <= '1';
79                     else
80                         bitcnt <= bitcnt - 1;
81                     end if;
82                 else
83                     baudratecnt <= baudratecnt - 1;
84                 end if;
85             end case;
86         end if;
87     end if;
88 end process;
89 end architecture Behavioral;

```

Listing A.6: Vollständiger VHDL-Quelltext des VME-Submoduls. Dieses Modul stellt später die Verbindung zwischen Hauptmodul und dem VME-Bus her.

```

1 -----
2 -- Company: RWTH Aachen – III. Physikalisches Institut A
3 -- Engineer: Carsten Heidemann
4 --

```

```

5  -- Design Name: Main VME Modul
6  -- Module Name: VME_interface – Behavioral
7  -- Project Name: Logic Modules for VDC
8  -- Target Devices:
9  -- Tool versions:
10 -- Description: Interface for the VME communication with the three models
11 --
12 -- Dependencies: currently none
13 --
14 -- Revision:
15 -- Revision 0.2 – beta, has to be tested
16 -- Additional Comments:
17 --
18 -----
19 library IEEE;
20 USE IEEE.STD_LOGIC_1164.ALL;
21 USE IEEE.NUMERIC_STD.all;
22
23 ----- Uncomment the following library declaration if instantiating
24 ----- any Xilinx primitives in this code.
25 library UNISIM;
26 use UNISIM.VComponents.all;
27
28 entity vme_trans is
29   generic(syssspeed, baudrate : integer);
30   port(
31     -- Hardware Connections
32     FPGA_clk : in std_logic;
33     Bus_Reset : in std_logic;
34     Bus_AM : in std_logic_vector(5 downto 0);
35     Bus_Address_strobe : in std_logic;
36     Bus_Data_Strobe_0 : in std_logic;
37     Bus_Data_Strobe_1 : in std_logic;
38     Bus_Long_Word : in std_logic;
39     Bus_Write : in std_logic;
40     Bus_Data : inout std_logic_vector(31 downto 0);
41     Bus_Address : inout std_logic_vector(31 downto 0);
42     Bus_Error : out std_logic; -- data transfer not possible
43     Bus_Data_Ack : out std_logic; -- data is on data lines
44     Bus_Adr_Compare : in std_logic; -- Hardware Comparator for Bus_Address;
45     -- Interrupt Ports
46     Bus_IACK : in std_logic; -- wenn 1 -> kein IRQ
47     -- Communication to Main_Modul
48     copy_req : out std_logic;
49     lese_req : out std_logic;
50     schreib_req : out std_logic;

```

```

51     reg_adr : out unsigned(7 downto 0);
52     neu_reg_wert : out unsigned(63 downto 0);
53     aktuell_reg_wert : in unsigned(63 downto 0)
54 );
55 end entity vme_trans;
56 D0: IOBUF port map (I => DQ_OUT, O => DQ_IN, T => DQ_CONTROL, IO => Bus_Data(0)
57 );
58 architecture Behavioral of reg_trans is
59     constant maxcnt: natural := syssspeed / ???; -- min Breite der Pulse/ Strobos
60     subtype byte is unsigned(7 downto 0);
61     subtype halbbyte is unsigned(3 downto 0);
62     type statusType is (
63     ...
64     ...
65     );
66     signal status : statusType := xxxxxxxxxxxx ;
67     signal data_in : std_logic_vector(31 downto 0);
68     signal data_out : std_logic_vector(31 downto 0);
69     signal data_direction : std_logic;
70     -- Data Lines IN / OUT Separation
71     D0: IOBUF port map (I => data_out(0), O => data_in(0), T => data_direction, IO =>
72     Bus_Data(0));
73     D1: IOBUF port map (I => data_out(1), O => data_in(1), T => data_direction, IO =>
74     Bus_Data(1));
75     D2: IOBUF port map (I => data_out(2), O => data_in(2), T => data_direction, IO =>
76     Bus_Data(2));
77     D3: IOBUF port map (I => data_out(3), O => data_in(3), T => data_direction, IO =>
78     Bus_Data(3));
79     D4: IOBUF port map (I => data_out(4), O => data_in(4), T => data_direction, IO =>
80     Bus_Data(4));
81     D5: IOBUF port map (I => data_out(5), O => data_in(5), T => data_direction, IO =>
82     Bus_Data(5));
83     D6: IOBUF port map (I => data_out(6), O => data_in(6), T => data_direction, IO =>
84     Bus_Data(6));
85     D7: IOBUF port map (I => data_out(7), O => data_in(7), T => data_direction, IO =>
86     Bus_Data(7));
87     D8: IOBUF port map (I => data_out(8), O => data_in(8), T => data_direction, IO =>
88     Bus_Data(8));
89     D9: IOBUF port map (I => data_out(9), O => data_in(9), T => data_direction, IO =>
90     Bus_Data(9));
91     D10: IOBUF port map (I => data_out(10), O => data_in(10), T => data_direction, IO =>
92     Bus_Data(10));
93     D11: IOBUF port map (I => data_out(11), O => data_in(11), T => data_direction, IO =>
94     Bus_Data(11));

```

```
83 D12: IOBUF port map (I => data_out(12), O => data_in(12), T => data_direction, IO =>
    Bus_Data(12));
84 D13: IOBUF port map (I => data_out(13), O => data_in(13), T => data_direction, IO =>
    Bus_Data(13));
85 D14: IOBUF port map (I => data_out(14), O => data_in(14), T => data_direction, IO =>
    Bus_Data(14));
86 D15: IOBUF port map (I => data_out(15), O => data_in(15), T => data_direction, IO =>
    Bus_Data(15));
87 D16: IOBUF port map (I => data_out(16), O => data_in(16), T => data_direction, IO =>
    Bus_Data(16));
88 D17: IOBUF port map (I => data_out(17), O => data_in(17), T => data_direction, IO =>
    Bus_Data(17));
89 D18: IOBUF port map (I => data_out(18), O => data_in(18), T => data_direction, IO =>
    Bus_Data(18));
90 D19: IOBUF port map (I => data_out(19), O => data_in(19), T => data_direction, IO =>
    Bus_Data(19));
91 D20: IOBUF port map (I => data_out(20), O => data_in(20), T => data_direction, IO =>
    Bus_Data(20));
92 D21: IOBUF port map (I => data_out(21), O => data_in(21), T => data_direction, IO =>
    Bus_Data(21));
93 D22: IOBUF port map (I => data_out(22), O => data_in(22), T => data_direction, IO =>
    Bus_Data(22));
94 D23: IOBUF port map (I => data_out(23), O => data_in(23), T => data_direction, IO =>
    Bus_Data(23));
95 D24: IOBUF port map (I => data_out(24), O => data_in(24), T => data_direction, IO =>
    Bus_Data(24));
96 D25: IOBUF port map (I => data_out(25), O => data_in(25), T => data_direction, IO =>
    Bus_Data(25));
97 D26: IOBUF port map (I => data_out(26), O => data_in(26), T => data_direction, IO =>
    Bus_Data(26));
98 D27: IOBUF port map (I => data_out(27), O => data_in(27), T => data_direction, IO =>
    Bus_Data(27));
99 D28: IOBUF port map (I => data_out(28), O => data_in(28), T => data_direction, IO =>
    Bus_Data(28));
100 D29: IOBUF port map (I => data_out(29), O => data_in(29), T => data_direction, IO =>
    Bus_Data(29));
101 D30: IOBUF port map (I => data_out(30), O => data_in(30), T => data_direction, IO =>
    Bus_Data(30));
102 D31: IOBUF port map (I => data_out(31), O => data_in(31), T => data_direction, IO =>
    Bus_Data(31));
103 begin
104 end architecture Behavioral;
```