

JÜRGEN HORN

MESSUNGEN AN EINER
VIELDRAHTDRIFTKAMMER
UND
ENTWICKLUNG EINES VME-GRAPHIKSYSTEMS
FÜR DIE BESCHLEUNIGERSTEUERUNG

DIPLOMARBEIT

MAI 1991

**Messungen an einer
Vieldrahtdriftkammer
und
Entwicklung eines VME-Graphiksystems
für die Beschleunigersteuerung**

DIPLOMARBEIT

von

Jürgen Horn

Institut für Kernphysik

Technische Hochschule Darmstadt

Mai 1991

Inhaltsverzeichnis

Zusammenfassung	
1 Einleitung	1
2 Messungen an einer Vieldrahtdriftkammer	3
2.1 Überblick über Strahlerzeugung und Meßsystem	3
2.2 Physikalische Grundlagen zur Funktionsweise der Driftkammer	8
2.3 Aufbau der Driftkammer	11
2.4 Ausleseelektronik	14
2.5 Auswerteverfahren	14
2.6 Ausführung der Messungen	16
2.7 Auswertung und Diskussion der Ergebnisse	17
3 Entwicklung eines Graphiksystems für die Beschleunigersteuerung	22
3.1 Motivation und Zielvorstellungen	22
3.2 Hardwarebeschreibung	23
3.3 Grundlagen zur Software	26
3.4 Entwicklung der Systemsoftware	31
3.5 Integration in die Applikationsumgebung	32
3.6 Funktionsbeschreibung	33
3.7 Test	33
3.8 Weitere Entwicklungsmöglichkeiten Software	34
3.9 Weitere Entwicklungsmöglichkeiten Hardware	34
4 Status und Schlußbemerkung	36
Anhang	37
A Dokumentation Terminalemulation	37
A.1 ANSI/VT Modus	37
A.2 Tektronix 4014/Retrographics 640 Modus	40
A.3 Nonstandard Funktionen	42
B Dokumentation Graphiksystemsoftware	44
B.1 Kontrollsequenzen der GKS-Anbindung	44
B.2 Parameterkodierung	44
B.3 Erklärung der GKS-Kommandos	44

C Dokumentation GKS-Treiber	50
C.1 Modulstruktur	50
C.2 Gerätebeschreibungstabelle	50
 Literaturverzeichnis	 52

Zusammenfassung

Der neue Darmstädter supraleitende Elektronenbeschleuniger S-DALINAC im Institut für Kernphysik der TH Darmstadt erlaubt aufgrund seiner technischen Daten neuartige Experimente. So ermöglicht zum Beispiel der Dauerstrichelektronenstrahl die Durchführung von Koinzidenzexperimenten. Die verbesserte Energieauflösung erhöht die Aussagekraft der Ergebnisse. Die dadurch gestiegenen Anforderungen machten ein neues Meßsystem notwendig. Daher wurde ein Spektrometer nach modernen Vorschlägen gebaut. Seine Konstruktion besitzt mehrere bauliche Besonderheiten. Das Spektrometer wird aufgrund seiner Form als QClam (Q = Quadrupol, Clam = Muschel) bezeichnet. Die Elektronen werden von drei Vieldrahtdriftkammern erfaßt, mit denen es möglich ist, den Richtungvektor des Elektrons zu bestimmen. Durch Rückrechnung erhält man den Impuls und den Richtungsvektor des Elektrons auf dem Target. Dadurch wird eine Rückstoßkorrektur möglich.

Im Rahmen des ersten Teils dieser Diplomarbeit wurde die Funktionstüchtigkeit der Vieldrahtkammern nachgeprüft. Die Messungen mit einer ^{90}Sr -Quelle ergaben die theoretisch zu erwartenden Werte für Zeit- und Ortsauflösung. Die Zeitauflösung erreicht einen Wert von 7 ns. Die sich daraus ergebende Ortsauflösung je Draht beträgt 0.4 mm. Diese Werte werden durch die für kleine Energien stark einfließende Kleinwinkelstreuung bestimmt. Durch weitergehende Auswertung wurden die Bahnverläufe der Elektronen errechnet, wodurch es möglich wurde, die Lage und Größe der Elektronenquelle zu bestimmen und somit die Konsistenz der Meßdaten nachzuweisen.

Im zweiten Teil der Diplomarbeit wurde ein schnelles Farbgraphiksystem für die Steuerung des neuen Beschleunigers entwickelt. Der supraleitende Beschleuniger wird mit einem Steuersystem auf Basis mehrerer vernetzter Computer bedient. Die Steuerrechner, die mit der Beschleunigerhardware in Verbindung stehen, werden dazu von einem übergeordneten Bedienrechner kontrolliert. Dieser Bedienrechner besitzt eine moderne graphische Benutzeroberfläche. Der Benutzer erhält die Möglichkeit, übersichtlich alle Elemente des Beschleunigers darzustellen und zu beeinflussen, sowie Zustandmeldungen mehrerer Geräte anzuzeigen. Diese graphische Benutzeroberfläche läuft auf dem entwickelten Graphiksystem.

Die Systemsoftware besteht aus einer Terminalemulation für ein VT640-Graphikterminal mit spezifischen Erweiterungen sowie aus einer GKS-Schnittstelle. Diese wird auf der VAX-Seite ergänzt durch einen GKS-Gerätetreiber für das am Institut verwendete GKS-Paket.

Die Software wurde auf breite Anwendungsfähigkeit hin optimiert und mit zahlreichen Graphikprogrammen erfolgreich getestet. In umfangreichen Tests wurde die fehlerfreie Funktion des gesamten Systems nachgewiesen. Die Verwendbarkeit für die Beschleunigersteuerung ist damit gesichert.

1 Einleitung

Seit fast 30 Jahren werden am Institut für Kernphysik der TH Darmstadt elektromagnetische Kernübergänge mittels hochauflösender, inelastischer Elektronenstreuung bei niedrigem Impulsübertrag untersucht. Dabei erzielte man mit dem Studium der elektrischen und magnetischen Multipolresonanzen große Erfolge. Bis Ende 1988 wurde dazu der gepulste Linearbeschleuniger DALINAC [1] verwendet, der eine Maximalenergie von 70 MeV bei einem mittleren Strahlstrom von ca. $10 \mu\text{A}$ lieferte. Bedingt durch das niedrige Tastverhältnis von $7.5 \cdot 10^{-4}$ konnten mit dem DALINAC nur Einarmexperimente des Typs (e, e') durchgeführt werden, d.h. lediglich die gestreuten Elektronen wurden spektroskopiert.

Zum besseren Verständnis der Natur der Kernkräfte und des Kernaufbaus benötigt man jedoch Experimente der Art $(e, e'x)$, die den Nachweis von weiteren, aus dem hochangeregten Kern emittierten Teilchen (z.B. $x = \gamma, p, n, \dots, f$), in Koinzidenz mit den gestreuten Elektronen gestatten. Für solche Experimente eignet sich am besten ein kontinuierlicher Elektronenstrahl, da mit ihm das Verhältnis von zufälligen zu echten Koinzidenzen optimal wird.

Der supraleitende, rezirkulierende Linearbeschleuniger S-DALINAC [2], der seit 1982 im Aufbau ist und dessen supraleitende Injektorstrukturen erstmals im August 1987 erfolgreich betrieben werden konnten [3, 4], liefert einen solchen kontinuierlichen Elektronenstrahl (Continuous Wave, CW). Im Dezember 1990 wurden die Rezirkulationen vollständig in Betrieb genommen, wodurch eine Energie von bis zu 75 MeV erreicht werden konnte. Der Beschleuniger ist ausgelegt für eine Endenergie von 130 MeV, einen Strahlstrom von $20 \mu\text{A}$, und eine Energieschärfe von 10^{-4} . Die hohe Energieschärfe und die Emittanz des Elektronenstrahls ermöglichen es, Grundlagenforschung unter anderem auch auf dem Gebiet des Freien Elektronen Lasers (FEL) zu betreiben [5].

Um die oben beschriebenen Koinzidenzmessungen zu ermöglichen, wurde ein neues Spektrometer erforderlich, das auch die verbesserten Parameter des neuen Beschleunigers ausnutzen kann. Zu diesem Zweck wurde das QClam-Spektrometer aufgebaut [6]. Es zeichnet sich durch einen großen aktiven Raumwinkel und hohe Impulsakzeptanz aus. Zur Detektion der Elektronen werden drei Vieldrahtdriftkammern [7] verwendet. Diese werden elektronisch ausgelesen, durch ein Multiprozessorsystem [8] ausgewertet und durch ein CAMAC-Subsystem an ein übergeordnetes Meßdatenerfassungssystem (GOOSY) weitergeleitet [9].

Im Rahmen dieser Arbeit wurden Testmessungen an den Vieldrahtdriftkammern ausgeführt, um deren Funktionstüchtigkeit zu zeigen und um die Parameter zu bestimmen, die die Auflösung und die Qualität des Meßsystems beeinflussen. Dazu wurden die gewonnenen Meßdaten mit Hilfe eigenentwickelter Analyseprogramme ausgewertet.

Die Unterstützung der Steuerung des neuen supraleitenden Beschleunigers war die Motivation für den zweiten Teil dieser Arbeit. Die große Zahl von Dipolen und Quadrupolen, Strahlkorrekturspulen, Linsen und anderen Elementen, aus der sich die Strahlführung zusammensetzt, erfordert ein komplexes Steuerungs- und Kontrollsystem. Dieses ist mit Hilfe eines rechnergestützten Steuersystems auf Basis eines PDP-11 Prozeßrechners implementiert, wodurch sich eine relativ einfache und übersichtli-

che, aber auch in Hinblick auf zukünftige Erweiterungen flexible Lösung ergibt. Der Prozeßrechner ist im Beschleunigerlabor installiert. Die lokale Bedienung des Rechners, und damit auch der an ihn angeschlossenen Geräte, erfolgt über ein Touch-Terminal und mit speziellen Drehknöpfen.

Zur Bedienung des Beschleunigers vom etwa 50 m entfernten Kontrollraum wurde ein verteiltes Bediensystem entwickelt. Die Bedienerführung geschieht dabei mit einer graphikorientierten Benutzeroberfläche. Das Kontrollprogramm [10] läuft im lokalen VAX-Cluster der Beschleunigergruppe und verwendet das genormte Graphische Kern System (GKS) zur Visualisierung.

Im Rahmen dieser Arbeit wurde für diese Anwendung ein schnelles Farbgraphiksystem entwickelt, das den hohen Anforderungen an Auflösung, Geschwindigkeit, und Darstellungsqualität entspricht. Dazu wurde ein auf dem VME-Bus basierendes modulares System verwendet, das neben der Zentraleinheit eine Graphik- und eine Ethernetkarte umfaßt. Alle VME-Module wurden selbst aufgebaut, angepasst und getestet.

Die Arbeit gliedert sich in zwei Hauptteile. Im ersten Teil werden die Messungen an der Vieldrahtdriftkammer beschrieben. Hierzu wird erst ein Überblick über das Meßsystem sowie eine Beschreibung von Aufbau und Funktionsweise der Driftkammer gegeben. Daran schließt sich eine Beschreibung der Auswerteverfahren an. Eine detaillierte Darstellung der Messungen und die Diskussion der Ergebnisse schließt diesen Teil ab.

Im zweiten Teil wird die Entwicklung des Graphiksystems für die Beschleunigersteuerung beschrieben. Nach einer Diskussion der Zielvorstellungen, die zur Wahl eines solchen Systems führten, werden die Anforderungen an die Soft- und Hardware beschrieben. Dem folgt die Beschreibung der Hardwarebausteine mit einer Darstellung der Änderungen, die sich im Laufe des Aufbaus als notwendig erwiesen. Die Beschreibung der Software umfasst eine Darstellung der Terminalemulation, der wesentlichen Merkmale von GKS in Bezug auf die Anwender- und auf die Gerätetreiberschnittstelle, der verwendeten Fensterdarstellungstechnik und der Kommunikation zwischen den Systemen. Die Erfahrungen bei der Entwicklung der Software und der Integration in die Anwendungsumgebung sowie die verwendete Teststrategie zur Verifizierung der Systemsoftware werden im Anschluß behandelt. Als Abschluß dieses Teils wird auf Erweiterungsmöglichkeiten des Graphiksystems eingegangen.

Als Anhang wird eine Dokumentation der Software für die Terminalemulation sowie des GKS-Treibers gegeben. Darin enthalten ist eine vollständige Beschreibung aller implementierten Funktionen.

2 Messungen an einer Vieldrahtdriftkammer

In diesem Abschnitt werden die Testmessungen an der Vieldrahtdriftkammer beschrieben. Zuvor wird ein Überblick über das Meßsystem und die grundsätzliche Funktionsweise von Driftkammern gegeben.

2.1 Überblick über Strahlerzeugung und Meßsystem

Der neue Darmstädter supraleitende Beschleuniger S-DALINAC besitzt im Vergleich zum bisherigen Beschleuniger viele Besonderheiten, Verbesserungen und Neuheiten. Die wichtigsten neuen Eigenschaften sind:

- Das Tastverhältnis beträgt 1:1. Diese Betriebsart wird Dauerstrichbetrieb (continuous wave, cw) genannt. Das bedeutet, daß der Elektronenstrahl kontinuierlich eingeschaltet und, abgesehen von der durch die Hochfrequenz bedingten Mikrostruktur, nicht gepulst ist. Daraus folgt unter anderem, daß Koinzidenzexperimente möglich werden.
- Die maximal erreichbare Energie beträgt 130 MeV, der maximale Strahlstrom $20 \mu\text{A}$. Diese im Vergleich zum bis 1988 betriebenen DALINAC etwa verdoppelte Endpunktenergie erweitert den nutzbaren Energiebereich und ermöglicht neue, interessante Experimente.
- Die Energieschärfe beträgt 10^{-4} bei 130 MeV. Das bedeutet, daß die Energie der Elektronen nur sehr wenig um ihren Mittelwert streut. Daraus folgt für die

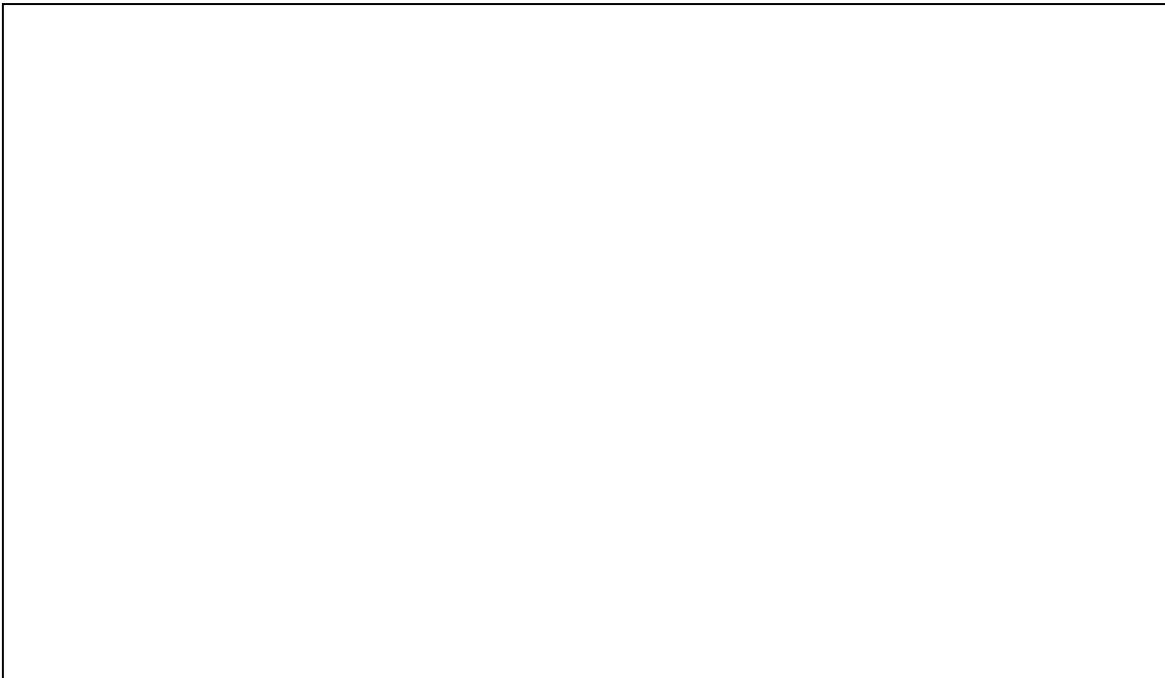


Abb. 2.1: Überblick über den Beschleuniger.

Experimente eine Verbesserung der Auflösung. So können auch noch sehr eng benachbarte Zustände getrennt werden.

Die wesentlichen Elemente des supraleitenden Beschleunigers sind in Abb. 2.1 [2] dargestellt. Unmittelbar hinter der Elektronenkanone wird der Elektronenstrahl elektrostatisch auf eine Energie von 270 keV beschleunigt. In der sich anschließenden Chopper-Prebunchersektion wird dem Strahl die zur Hochfrequenzbeschleunigung erforderliche Mikrostruktur, bestimmt durch die Länge der mit der Beschleunigerfrequenz erzeugten Bunche, aufgeprägt. Der Injektor [11], der aus einer fünfzelligen, als Frequenz- und Phasenreferenz dienenden Einfangsektion und zwei zwanzigzelligen Beschleunigungsstrukturen aufgebaut ist, liefert eine Maximalenergie von 11 MeV. Im Anschluß an den Injektor befinden sich zwei Experimentierplätze zur Untersuchung von Kernresonanzfluoreszenz [12] und Elektronen-Channeling [13].

Nach der Vorbeschleunigung wird der Strahl mittels eines magnetischen Umlenk-systems in den Hauptbeschleuniger eingeschossen. Mit dessen acht zwanzigzelligen Beschleunigungsstrukturen lassen sich maximal 40 MeV Energiegewinn pro Durchlauf erzielen. An den Hauptbeschleuniger schließen sich zwei getrennte, isochrone Rezirkulationswege [14, 15, 16] an, die es ermöglichen, den Hauptbeschleuniger dreifach zu durchlaufen. Je nach Anzahl der betriebenen Rezirkulationen kann der Strahl so bei Energien bis 50, 90, oder 130 MeV mit Hilfe der Extraktion aus dem S-DALINAC ausgekoppelt und in das Strahlführungs- und Energieanalysesystem geleitet werden, das sich in der Experimentierhalle befindet. Die gesamte Strahlführung in der Beschleuniger- und der Experimentierhalle ist in Abb. 2.2 [17] gezeigt.

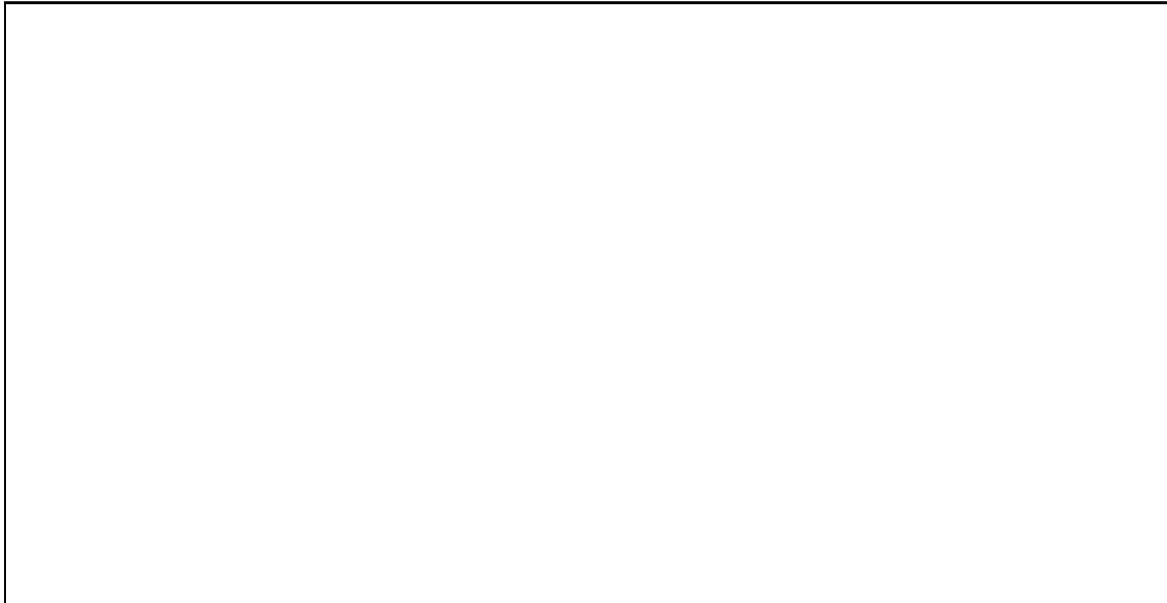


Abb. 2.2: Überblick über die Strahlführung. In der Experimentierhalle wird der Strahl in das 169°-Spektrometer (unten links), in das neue QClam-Spektrometer (rechts) oder in einen Channeling-Meßplatz (oben rechts) geleitet. Zwischen Hauptbeschleuniger und erster Rezirkulation ist der im Aufbau befindliche Freie Elektronen Laser zu erkennen.

Um die besseren Eigenschaften des Beschleunigers zu nutzen, reichten die bestehenden Meßeinrichtungen nicht aus. Das alte 169° -Spektrometer hat aufgrund seines geringen Raumwinkels für Koinzidenzmessungen eine zu geringe Koinzidenznachweiswahrscheinlichkeit. Daher wurde ein neues Spektrometer geplant und aufgebaut, das den Parametern des Beschleunigers Rechnung trägt sowie den koinzidenten Nachweis von Elektronen und Hadronen erlaubt. Das QClam-Spektrometer zeichnet sich durch folgende Eigenschaften aus:

- Der aktive Raumwinkel beträgt 40 msr.
- Die Impulsakzeptanz $\Delta p/p$ beträgt bis zu 20 %. Es können also gleichzeitig Elektronen mit bis zu 20 % unterschiedlichem Impuls nachgewiesen werden.
- Die Energieauflösung $\Delta E/E$ erreicht einen Wert von 10^{-4} und entspricht damit der Energieschärfe des neuen Beschleunigers.

Die vom Target kommenden Elektronen werden im Magnetspektrometer zu den Elektronendetektoren abgelenkt. Die Hadronen werden durch rund um das Target angebrachte Halbleiterdetektoren nachgewiesen. Aus den oben genannten Daten für das Spektrometer ergeben sich folgende Anforderungen an den Elektronendetektor:

- Die Ortsauflösung muß besser als 0.2 mm sein, um die gewünschte Energieauflösung zu erreichen.
- Für die Winkelauflösung muß ein Wert von 0.5° verlangt werden. Damit erhält man die Daten für die genaue Bestimmung des Elektronenvektors zur Rückrechnung auf das Target.
- Der Aufbau soll einfach und robust sein, um eine hohe Zuverlässigkeit zu gewährleisten.
- Die Justierung muß einfach und reproduzierbar gestaltet werden, damit die Meßergebnisse aussagefähig sind.
- Zur Verarbeitung hoher Zählraten muß das gesamte Auslesesystem schnell und leistungsfähig sein.

Um diesen Anforderungen gerecht zu werden, wurden die Elektronendetektoren als ein Paket aus 3 vertikalen Driftkammern zur Bahnvektorbestimmung, sowie einem Szintillator und einem Čerenkov-Detektor zur Bestimmung des Zeitpunkts des Elektronendurchgangs gebaut. Der Elektronenstrahl wird, von der Streukammer kommend, durch das energieanalysierende Magnetsystem nach oben abgelenkt. Im Bereich der Fokalebene des Magneten durchlaufen die Elektronen die Driftkammern, dann den Szintillator und den Čerenkov-Detektor. Die Fokalebene des Spektrometers ist die Ebene, in der alle Elektronen, die an einem bestimmten Ort mit einem bestimmten Impulsvektor in das Spektrometer eingetreten sind, auf einen Punkt abgebildet werden. Eine Seitenansicht des Spektrometers ist in Abb. 2.3 [18], Details des Elektronendetektors sind in Abb. 2.4 dargestellt.

Streuwinkel und Energie der in das Spektrometer gestreuten Elektronen fließen in die (x,y)-Koordinate der austretenden Elektronen in der Fokalebene ein. Zwei Driftkammern besitzen senkrecht zur Dispersionsrichtung gespannte Drähte und liefern die x-Koordinate, die der Energieaufspaltung des Spektrometernagnetens entspricht. Zwei

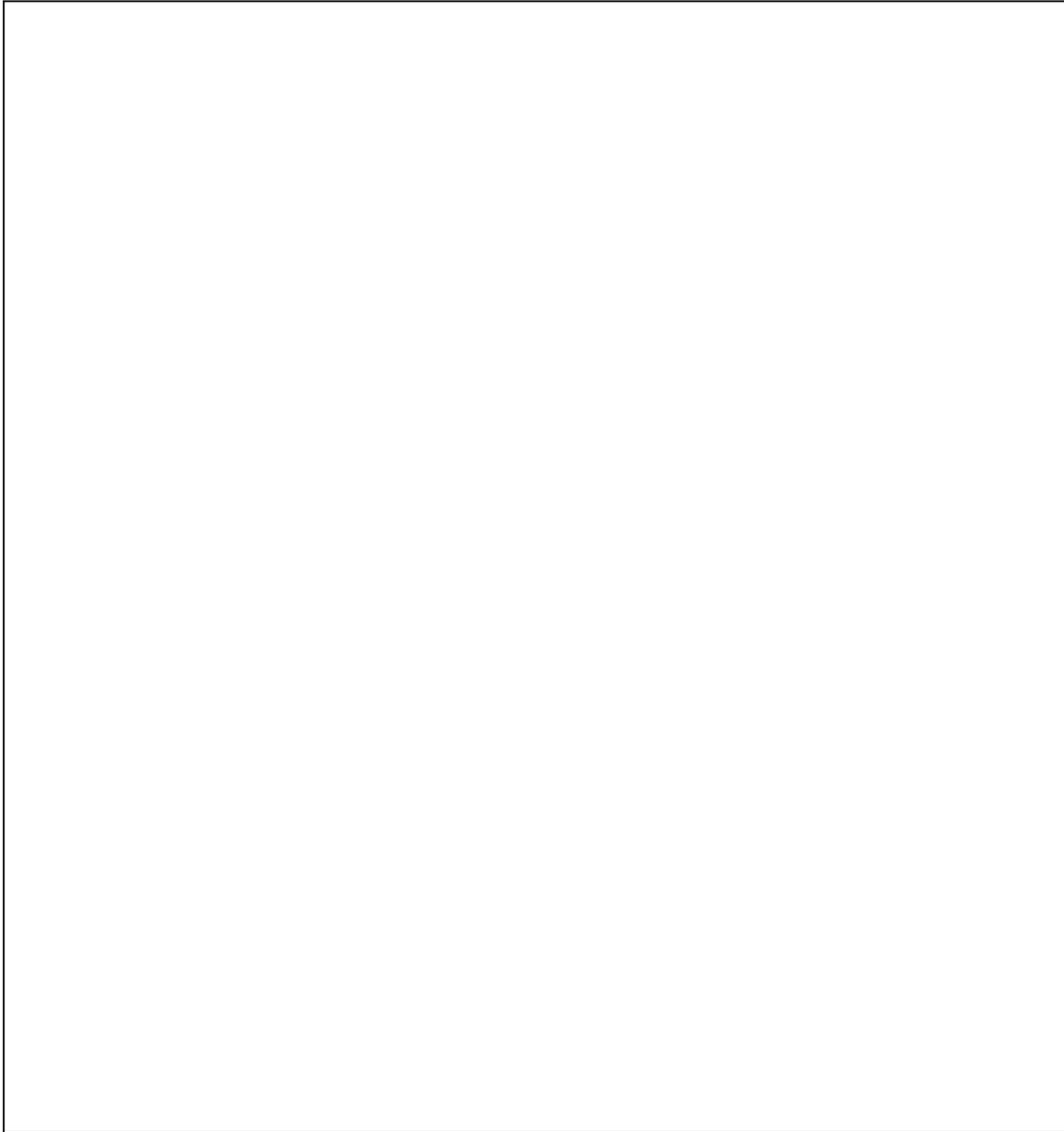


Abb. 2.3: Seitenansicht des QClam-Spektrometers. Unten links befindet sich die Streukammer mit der darauf montierten Targetleiter. Die gestreuten Elektronen gelangen in das QClam-Magnetsystem und werden dort nach oben abgelenkt. Dort sitzen zur Detektion der Elektronen in der Fokalebene die drei Driftkammern, der Szintillator und der Čerenkov-Detektor. Das gesamte Spektrometer kann um die senkrechte Mittelachse der Streukammer gedreht werden.

x-Driftkammern wurden gewählt, um die Genauigkeit der Bahnbestimmung zu verbessern, und um den Winkel zu bestimmen, mit dem das Elektron durch das Kammerpaket geflogen ist. Um die y-Koordinate messen zu können, besitzt die dritte Driftkammer in einem Winkel von 26.5° relativ zu den Drähten der anderen Kammern angebrachte

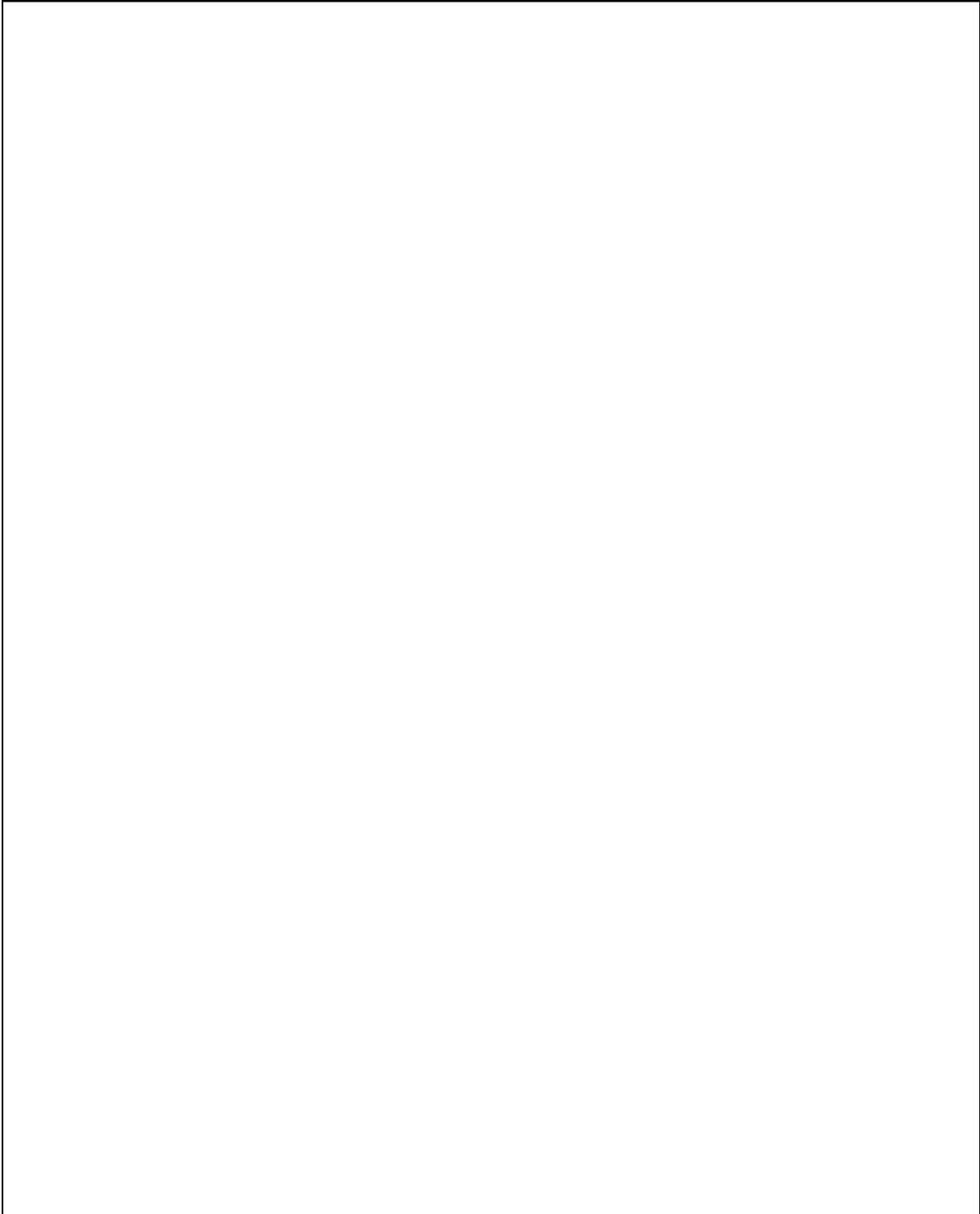


Abb. 2.4: Detaildarstellung des Detektorsystems für den Elektronennachweis.

Zähldrähte. Dieser Winkel wurde gewählt, da bei genau orthogonal zu den Drähten der anderen Kammern gespannten Drähten die freie Drahtlänge zu groß werden würde. Dadurch würden die Kräfte des Hochspannungsfeldes auf die Drähte zu einer starken Ortsabweichung und somit zu einer wesentlichen Beeinträchtigung der Genauigkeit führen. Der Winkel von 26.5° erleichtert die Auswertung dadurch, daß sein Tangens gerade 0.5 ergibt. Zusammen mit den Daten der x-Driftkammern eignet sich diese Anordnung zur Bestimmung der y-Koordinate. Aus den Durchstoßpunkten der drei Driftkammern kann der Richtungsvektor des Elektrons berechnet werden.

Der Szintillator liefert über zwei Photomultiplier den Zeitpunkt, an dem das Elektron die Fokalebene durchlaufen hat. Dieser Zeitpunkt ist als Anfangszeitpunkt des Driftvorgangs zur Bestimmung der Driftzeiten notwendig. Die Zusammenschaltung der beiden, am oberen und unteren Ende des Szintillators angebrachten Photomultiplier über eine Zeitmittelungsschaltung (Mean-Timer) sorgt für eine vom Ort des Elektronendurchgangs unabhängige Zeitreferenz.

Durch geeignete Zusammenschaltung von Koinzidenzen werden Störungen und Untergrundereignisse aussortiert und somit der Untergrund verringert. Die Verwendung sowohl von Szintillator als auch von Čerenkov-Detektor ist nötig, um die geringe Neutronenansprehwahrscheinlichkeit des Čerenkov-Detektors mit der geringen Gammaansprehwahrscheinlichkeit des Szintillators zu kombinieren. Diese beiden Ereignisarten sind nicht erwünscht.

2.2 Physikalische Grundlagen zur Funktionsweise der Driftkammer

Um die Funktionsweise von Driftkammern verstehen zu können, werden erst einige Grundlagen zur Ionisation, zur Drift geladener Teilchen in einem elektrischen Feld, und zur Gasverstärkung behandelt. Driftkammern funktionieren prinzipiell wie ein Proportionalzählrohr. Ein durch die mit Gas gefüllte Kammer fliegendes geladenes Teilchen ionisiert entlang seiner Flugbahn Gasteilchen und verliert dabei Energie. Die dabei entstehenden sekundären Elektronen und Ionen driften im elektrischen Feld zu den entsprechenden entgegengesetzt geladenen Elektroden. In der Nähe des positiv geladenen Drahtes lösen die Elektronen durch die stark ansteigende Feldstärke eine Ladungslawine aus, die bei Auftreffen auf den Draht einen Impuls erzeugt.

2.2.1 Ionisation

Der Energieverlust geladener Teilchen durch Ionisation beim Durchgang durch Materie wird allgemein durch die Bethe-Bloch-Gleichung (2.1) angenähert [19]. Sie beschreibt den differentiellen Energieverlust dE/dx des Teilchens durch Integration über die übertragene Energie E von einem mittleren Ionisationspotential I bis zur maximal möglichen Energie eines gestoßenen Elektrons ($2m_e c^2 \beta^2 / (1 - \beta^2)$).

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

Hierbei ist N_A die Avogadro-Konstante, Z , A und ρ die Ordnungszahl, Massenzahl, und Dichte des Materials, z und β die Ladungszahl und der relativistische Geschwindigkeitsfaktor des Projektils. m_e und r_e ist die Elektronenmasse und der klassische Elektronenradius. Der Faktor $4\pi N_A r_e^2 m_e c^2$ beträgt $0.3071 \text{ MeVcm}^2\text{g}^{-1}$. I ist das effektive Ionisationspotential des Materials. Dieses kann durch $I = I_0 Z$ angenähert werden, wobei der Faktor $I_0 = 12 \text{ eV}$ ist. Der differentielle Energieverlust hängt nur von der Geschwindigkeit $v = \beta c$ ab. Mit wachsendem β nimmt dE/dx erst mit $1/\beta^2$ auf ein Minimum ab, um dann wieder zuzunehmen. Für relativistische Elektronen kann fast immer der Fall annähernd minimaler Ionisierung angenommen werden.

2.2.2 Elektrisches Feld

Die Hochspannungsfolien werden gegenüber den Drähten auf ein negatives Potential von etwa 6 bis 10 kV gebracht. Dadurch entsteht zwischen Folien und Drahtebene ein elektrisches Feld. Da der Abstand der Drähte voneinander klein gegen die Entfernung der Drahtebene zu den Hochspannungsfolien ist, verläuft das Feld über weite Bereiche homogen. Potential Φ und Feldstärke E werden mit den folgenden Gleichungen beschrieben ([20],[21]):

$$U(x, y) = \frac{CU_0}{4\pi\epsilon_0} \cdot \left(\frac{2\pi l}{d} - \ln\left(4\left(\sin^2\left(\frac{\pi x}{d}\right) + \sinh^2\left(\frac{\pi y}{d}\right)\right)\right) \right) \quad (2.2)$$

$$E(x, y) = \frac{CU_0}{2\epsilon_0 d} \cdot \sqrt{\frac{1 + \tan^2\left(\frac{\pi x}{d}\right) \cdot \tanh^2\left(\frac{\pi y}{d}\right)}{\tan^2\left(\frac{\pi x}{d}\right) + \tanh^2\left(\frac{\pi y}{d}\right)}} \quad (2.3)$$

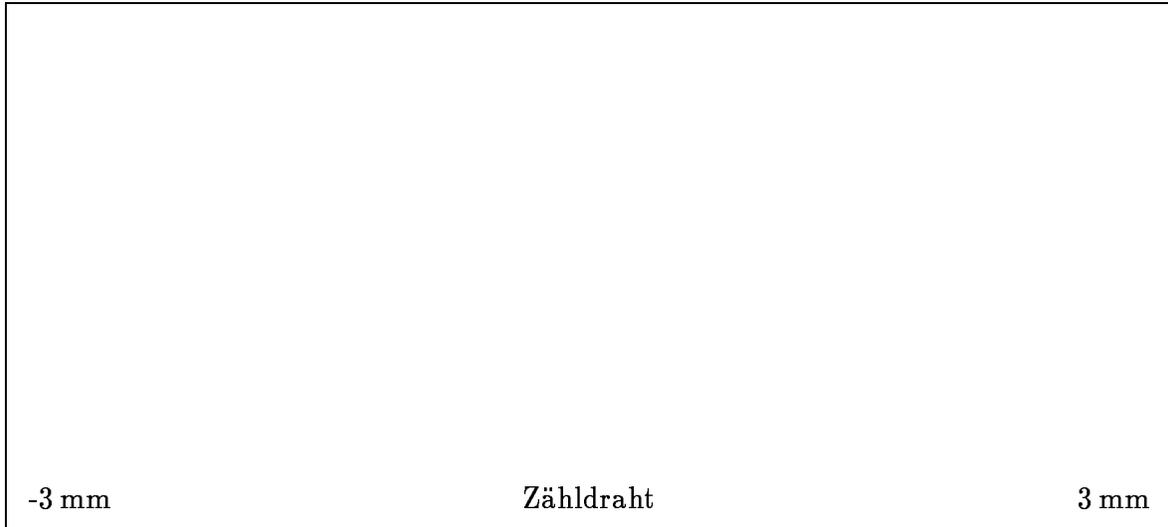


Abb. 2.5: Elektrisches Feld in einer Driftzelle. Der gezeigte Ausschnitt hat eine Größe von 2 mal 6 mm. Der Zähl draht befindet sich in der Mitte des Bildes, die Hochspannungsfolien links und rechts in vierfach größerem als dem dargestellten Abstand (12 mm). Man erkennt den homogene Feldbereich in weiten Bereichen der Kammer und das stark ansteigende Feld in unmittelbarer Drahtnähe.

Die Kapazität pro Länge C ist gegeben durch:

$$C = \frac{2\pi\epsilon_0}{\frac{\pi l}{d} - \ln\left(\frac{2\pi a}{d}\right)} \quad (2.4)$$

Dabei ist d der Abstand der Drähte voneinander, l der Abstand der Drahtebene zur Hochspannungsfolie und a der Drahtdurchmesser. Der mit Gleichung 2.3 berechnete Verlauf der Feldstärke ist in Abb. 2.5 dargestellt.

2.2.3 Gasverstärkung

Nur in Drahtnähe ist ein sprunghaftes Ansteigen der Feldstärke zu verzeichnen. Ab einer gewissen Feldstärke erhalten die driftenden Elektronen einen hohen Energiezuwachs, so daß sie weitere Zählgasatome ionisieren können. Daher werden unmittelbar in Drahtnähe lawinenartig ansteigend viele sekundäre Elektronen erzeugt. Da die Impulshöhe proportional zur Zahl der auf den Draht treffenden Ladungsträger ist, entsteht nur so ein meßbarer Impuls. Die Homogenität der Feldstärke im Driftbereich der Kammer wird durch je zwei Felddrähte zwischen den Zähldrähten verbessert, die den Abstand der Drähte auf ein Drittel verringern. Diese Felddhomogenisierungsdrähte haben einen größeren Durchmesser als die Zähldrähte. Dadurch ist die Feldstärke in der Nähe der Felddrähte wesentlich niedriger. Der beschriebene Lawineneffekt tritt bei den Felddrähten daher nicht auf.

Die in der Nähe der Zähldrähte entstehenden δ -Elektronen erzeugen hochenergetische Photonen, die beim Auftreffen auf die Kathodenfolie weitere Elektronen auslösen. Dies kann zu einer andauernden Entladung in der Kammer führen und somit Messungen unmöglich machen, bzw. sogar die Kammer zerstören. Um diesen Effekt zuverlässig zu verhindern, verwendet man im allgemeinen ein Gasgemisch aus dem Zählgas und einem mehratomigen Löschgas (quenching gas), das nicht ionisiert wird und mit seinen vielen Freiheitsgraden die Photonen absorbiert. In der hier beschriebenen Kammer wird ein Gasgemisch aus Argon und Isobutan verwendet. Die Gasverstärkung und die Driftgeschwindigkeit kann durch Veränderung der beiden Anteile in gewissen Grenzen an die Hochspannung angepasst werden.

2.2.4 Driftgeschwindigkeit

Die Drift der Elektronen wird durch die Kraft des elektrischen Feldes ausgelöst. Durch die Stöße gegen das in der Kammer vorhandene Gas entsteht ein mit der Geschwindigkeit wachsender Widerstand, so daß sich eine mittlere Driftgeschwindigkeit einstellt. Diese Driftgeschwindigkeit ist vom Druck, der Gasart, und der Hochspannung abhängig. Jedoch kann bei der Gasmischung Argon/Isobutan für eine Feldstärke ab etwa 1500 V/cm und Umgebungsdruck eine recht stabile Driftgeschwindigkeit von 50 $\mu\text{m}/\text{ns}$ abschätzen, die auch bei Änderungen der Gasmischung nicht wesentlich schwankt.

2.3 Aufbau der Driftkammer

Die Driftkammer besteht aus zehn rechteckigen, je 6 mm starken Kunststoffplatten aus einem glasfaserverstärktem Epoxidharz (G10), das sich durch hohe Verwindungssteifheit und Hochspannungsfestigkeit auszeichnet. Zur Erhöhung der Stabilität sind die Platten auf einen Aluminiumrahmen aufgeschraubt. Im aktiven Bereich der Driftkammer sind in den Kunststoffplatten rechteckige Öffnungen von $100 \cdot 12$ cm ausgeschnitten. Von außen nach innen betrachtet, sind bei der X2-Kammer, auf je einer Platte befestigt, eine Gasabschlußfolie, eine Hochspannungsfolie, die Drahtebene und dann wieder eine Hochspannungsfolie und eine Gasabschlußfolie angebracht. Die Platten werden untereinander mit O-Ringen abgedichtet. X1- und Y-Ebene sind übereinander in einer Kammer untergebracht. Eine Aufsicht auf die Y-Driftkammer ist in Abb. 2.6, ein Schnitt durch den X1/Y-Stapel in Abb. 2.7 [22] dargestellt.

Um eine geringe Streuung der Elektronen zu erreichen, müssen die Folien möglichst dünn ausgelegt werden. Gleichzeitig muß aber noch auf ausreichende Festigkeit geachtet werden. Auch die Dichte gegen das Durchlassen von Wasserdampf ist zu berücksichtigen.

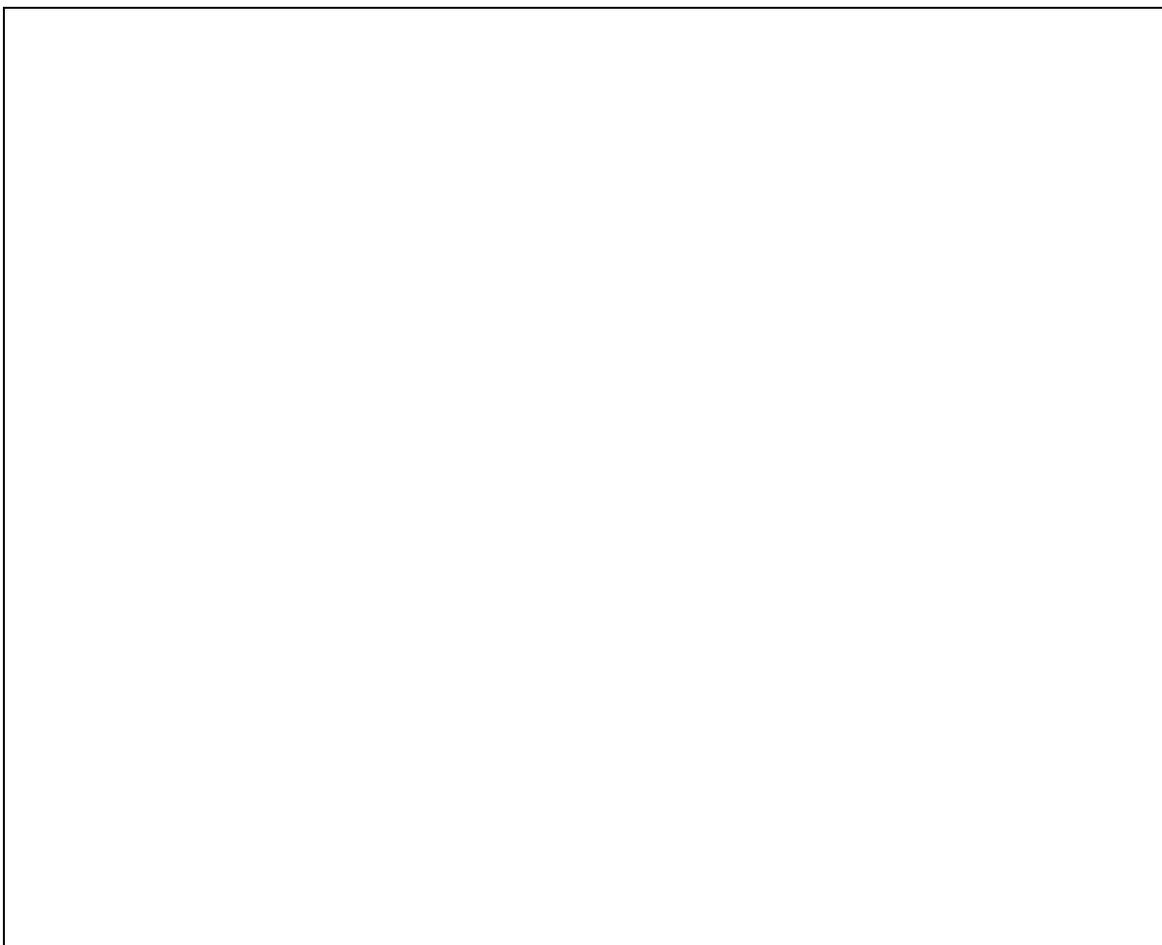


Abb. 2.6: Aufsicht auf die Y-Driftkammer. Zur besseren Übersicht ist nur jeder zehnte Draht eingezeichnet. Bei den X-Kammern sind die Zähldrähte senkrecht gespannt.

sichtigen. Zwischen diesen gegenläufigen Forderungen muß ein Kompromiß gefunden werden. Die Folien der ursprünglichen Driftkammer, mit der auch die Messungen ausgeführt wurden, waren insgesamt $16\ \mu\text{m}$ dick. Inzwischen werden jedoch Folien mit einer Gesamtstärke von $40\ \mu\text{m}$ eingesetzt, die die Kammer gegen ein Eindringen von Wasserdampf besser schützen [22].

Die Drähte sind im Abstand von 2 mm voneinander parallel gespannt, wobei zwischen zwei Zähldrähten (Durchmesser $20\ \mu\text{m}$) zwei Feldhomogenisierungsdrähte (Durchmesser $50\ \mu\text{m}$) liegen. Der Abstand zwischen zwei aktiven Drähten beträgt somit 6 mm. Daraus ergeben sich über die Gesamtlänge von 100 cm unter Abzug eines Randbereichs von je 2 cm 160 Zähldrähte. Diese werden nur bei der am weitesten vom Spektrometerausstritt angebrachten X2-Driftkammer voll genutzt, da hier die Aufweitung am größten ist. Bei den beiden anderen Driftkammern (X1 und Y) werden nur 128 Drähte ausgelesen. Die aktive Länge in Dispersionsrichtung (X) beträgt also bei der X1- und Y-Kammer 762 mm, bei der X2-Kammer 954 mm. Der Abstand der Drahtebene von den Hochspannungsfolien beträgt 12 mm.

Die Drähte sind herausgeführt und auf einer Anschlußplatine [23] befestigt, auf die die Vorverstärkerkarten gesetzt werden. Die Feldhomogenisierungsdrähte werden auf Massepotential gelegt. Durch Bohrungen am Rand der Platten wird die Kammer

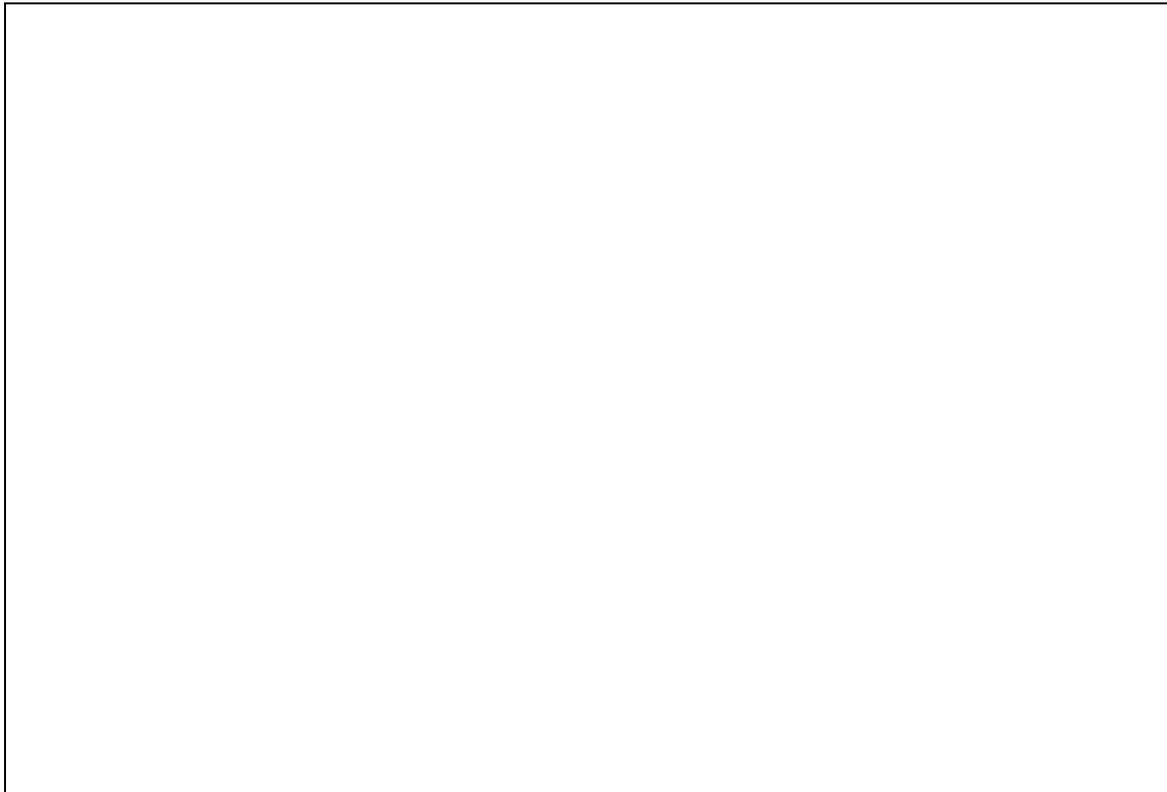


Abb. 2.7: Vertikaler Schnitt durch den Rand der X1/Y-Driftkammer, von der Seite aus gesehen. Die beiden aktiven Bereiche liegen zwischen den drei HV-Folien, deren Abstand je 24 mm beträgt. (GA – Gasabschlußfolie, HV – Hochspannungsfolie, D – Drahtebene).

mit Gas versorgt. Die Gasmischung und der Gasstrom wird durch einen elektronischen Durchflußregler kontrolliert und konstant gehalten. Ein Ausschnitt der Driftkammer mit den Driftzellen und einer typischen Elektronenbahn ist in Abbildung 2.8 zu sehen.



Abb. 2.8: Ausschnitt aus der aktiven Zone einer Driftkammer. Die Feldlinien sind nur für die aktiven Driftzellen eingetragen und durch einen radialen Bereich um den Zähl- draht und ansonsten homogenen Verlauf angenähert. Die gestrichelte Diagonale stellt die Bahn eines Elektrons dar, das die Kammer durchquert und 4 Drähte ansprechen läßt. Zwischen den Zähldrähten sind je zwei Feldhomogenisierungsdrähte zu erkennen

2.4 Ausleseelektronik

Die Ausleseelektronik [7, 8] kann anhand des Signalverlaufs beschrieben werden. Die auf einen Draht treffenden Sekundärelektronen lösen einen Impuls aus, der durch einen direkt an der Driftkammer sitzenden Vorverstärker verstärkt wird. Von dort führen, für jeden Draht getrennt, zwei differentielle Leitungen zu den im Meßrack eingebauten Diskriminatoren. Diese Diskriminatoren vergleichen die Impulshöhe mit einem von einem DA-Wandler gelieferten Vergleichswert, der je nach Erfordernissen auch während des Strahlbetriebs verändert werden kann. Wenn der Impuls mindestens genauso hoch ist wie dieser Vergleichswert, wird der Ausgang des Diskriminators gesetzt. Daraus resultiert eine Unterdrückung schwacher Störimpulse und von elektronischem Rauschen. Die weitere Bearbeitung des Signals geschieht rein digital. Um TDCs (Time to Digital Converter) zu sparen, wurden für die folgenden Stufen immer mehrere Drähte zusammengefaßt. Dies ist möglich, da ohnehin bei einem sinnvollen Ereignis aus geometrischen Gründen nur wenige Drähte (maximal 8 Drähte gleichzeitig) ansprechen können.

Ein gesetzter Diskriminatorexgang dient als Startsignal für einen TDC mit einer Auflösung von 1 ns. Dieser TDC wird von dem Signal, das von Szintillator und Čerenkov-Detektor kommt und um die maximale Driftzeit verzögert wird, gestoppt. Diese zeitlich umgekehrte Logik führt zu einer wesentlichen Vereinfachung des Schaltungsaufwands, da man so die sonst für jeden Draht notwendigen Verzögerungsbau- steine einspart.

Die so gewonnenen Driftzeiten werden über Lichtleiter an den dieser Kammer zugeordneten Ausleseprozessor gesendet. Dieser steht wie alle weiteren Einheiten im Meßraum und ist somit dauernd zugänglich. Dort kann nach ersten Plausibilitätsüberprüfungen der Durchstoßpunkt des Elektrons durch die Kammerebene bestimmt werden. Diese Werte werden über einen gemeinsam zugreifbaren Speicher an eine übergeordnete CPU geschickt. Diese überwacht die Vollständigkeit der Daten, d.h. daß alle drei Driftkammern ein Ereignis geliefert haben, sowie die Plausibilität der Werte. Wenn das Ereignis für sinnvoll gehalten wird, wird es mittels eines Lichtleiters an einen CAMAC-Prozessor gesendet. Dieser ist mit einer MicroVAX verbunden, auf der das Meßdatenerfassungsprogramm GOOSY läuft.

Mit GOOSY (GSI Online Offline SYstem) werden die Daten archiviert und ausgewertet. Es besteht aus mehreren Komponenten, die flexibel an Änderungen im Meßaufbau angepasst werden können. Beispielsweise kann während der Messung schon ein sich dauernd aktualisierendes Spektrum dargestellt werden. So können eventuelle Fehlein- stellungen rechtzeitig korrigiert werden.

2.5 Auswerteverfahren

2.5.1 Driftzeitenkonversion

Wie oben beschrieben, werden die Driftzeiten umgekehrt gemessen, d.h. vom eigentli- chen Stop-Impuls bis zum verzögerten Start-Impuls. Daher müssen sie erst noch umge- rechnet werden. Dazu muß der Meßwert von dem Verzögerungswert subtrahiert werden.

Der Verzögerungswert ist nicht für alle Drähte völlig gleich, da durch unterschiedliche Signallaufzeiten und Streuung der Elektronik ein nicht konstanter Verzögerungsanteil dazukommt. Um die genaue Verzögerungszeit für jeden Draht zu erhalten, muß ein unkonvertiertes Driftzeitspektrum aufgenommen werden. Die maximale Zeit, die man dabei sieht, entspricht der gesuchten Verzögerungszeit. Dieser Wert kann gut bestimmt werden, da an dieser Stelle das Driftzeitspektrum nach einer Überhöhung, die durch die Feldinhomogenität in Drahtnähe erzeugt wird, einen steilen Abfall zeigt.

2.5.2 Plausibilitätsprüfung

Zur Überprüfung der Plausibilität können unterschiedlich strenge Maßstäbe angelegt werden. Im Prinzip könnten noch Ereignisse, die nur aus zwei nicht nebeneinanderliegenden Drähten bestehen, ausgewertet werden. Dabei hat man allerdings keine Redundanz mehr. Daher wurden zum Test nur solche Ereignisse herangezogen, bei denen mindestens vier nebeneinanderliegende Drähte ohne Lücken angesprochen haben. Dann kann man durch die enthaltene Redundanz eine verbesserte Plausibilitätsprüfung durchführen, indem z.B. bei der linearen Regression der Korrelationskoeffizient überprüft wird. Folgende Ereignisse werden bei den hier ausgewerteten Messungen wegen mangelnder Plausibilität verworfen:

- Driftzeiten überschreiten die Grenzen, die durch die Kammergeometrie gegeben sind.
- Weniger als vier Drähte haben angesprochen.
- Lücken in den angesprochenen Drähten.
- Kleinste Driftzeit liegt am Rand.
- Driftzeitenkorrelation nicht ausreichend.

2.5.3 MIT-Algorithmus

Der MIT-Algorithmus ist ein vereinfachter Algorithmus zur Bestimmung des Durchstoßpunktes [24]. Er eignet sich durch seine wenigen einfachen Rechenoperationen zur Durchführung während der Messung (online). Dazu geht man wie folgt vor:

- Die angesprochenen Drähte werden auf ein Driftzeitminimum t_{min} hin untersucht.
- Von dem Draht mit der minimalen Driftzeit ausgehend werden die paarweise am weitesten außen liegenden angesprochenen Drähte mit den Driftzeiten t_1 und t_2 gesucht.
- Diese Driftzeiten verwendet man als Stützpunkte für eine Gerade durch die Drahtebene. Mit dem Durchgang dieser Gerade durch die Drahtebene ist der Durchstoßpunkt bestimmt.

Den Durchstoßpunkt durch die Drahtebene berechnet man mit der folgenden Formel:

$$x_d = x_{min} + \frac{d}{2} \cdot \frac{t_1 - t_2}{\max(t_1, t_2) - t_{min}} \quad (2.5)$$

Die Größe d ist der Abstand zwischen zwei Drähten, x_{min} der Ort des Drahtes mit der Driftzeit t_{min} .

2.5.4 Lineare Regression

Die Elektronenbahn kann auch durch Bestimmung einer Regressionsgeraden ermittelt werden. Dazu wird die kleinste Driftzeit wie oben ermittelt. Dann müssen die Driftzeiten auf einer Seite der kleinsten Driftzeit um den Nullpunkt gespiegelt werden. Dazu muß eine Richtung willkürlich festgelegt werden. Die so erhaltenen Driftzeiten werden dann mit linearer Regression analysiert. Daraus ergeben sich die Geradenparameter, unter anderem den Nulldurchgang, der dem Durchstoßpunkt entspricht. Der Korrelationskoeffizient kann zur Bestimmung des Fehlers der Driftzeiten für dieses Ereignisses herangezogen werden.

2.6 Ausführung der Messungen

Für die Testmessungen wurde die Kammer mit einer Hochspannung von 8 kV und einem Gasgemisch von 50 % Argon / 50 % Isobutan betrieben. Als Elektronenquelle wurde eine ^{90}Sr -Quelle mit einer Aktivität von 150 kBq verwendet. Sie wurde über der Kammer auf einer Kunststoffplatte installiert. In die Kunststoffplatte wurde ein Loch gebohrt, durch das die Quelle strahlen konnte. Durch diesen Aufbau ergibt sich eine in etwa halbkreisförmig gleichverteilt strahlende Punktquelle ohne Kollimation. Durch die relativ kleine aktive Zone der Kammer im Zusammenhang mit dem Abstand der Quelle werden daraus nur Elektronenbahnen mit einem bestimmten Winkelbereich selektiert. Daher ist bei der Bahnbestimmung nur ein relativ kleiner Winkelbereich für die Bahnen zu erwarten.

Da zum Zeitpunkt der Messungen wesentliche Teile des geschilderten Meßsystems noch nicht vorhanden waren, mußte ein abgeänderter Aufbau gewählt werden. So wurde nur eine Kammer untersucht, die mit nur einer Vorverstärkerkarte mit acht Kanälen ausgestattet war. Die von der Vorverstärkerkarte kommenden acht Leitungen wurden an ebenso viele Diskriminatoren angeschlossen. Die daran anschließende Zähler- und Steuerlogik war ebenfalls für alle acht Kanäle vorhanden. Die Steuerlogik sendete die Driftzeiten, die zu einem Ereignis gehören, über ein Lichtleiterinterface an den Auswerterechner. Dieser legte in seinem Speicher die zusammengehörenden Driftzeiten für alle Ereignisse ab, bei denen mindestens vier der acht Drähte angesprochen hatten. Dieses Kriterium wurde gewählt, da dadurch die Zahl der zwischenspeichernden Ereignisse stark verringert wurde und die Meßzeit so verlängert werden konnte. Außerdem ist es zur Auswertung in der Testphase sinnvoll, die Bahn des Elektrons aus möglichst vielen Punkten rekonstruieren zu können, da so eine Korrelationsanalyse möglich wird.

Die im Auswerterechner zwischengespeicherten Ereignisse wurden nach dem Ende der Messung über eine serielle Schnittstelle an einen VAX-Rechner geschickt und dort

in einer Datei festgehalten. Auf diese Datei können verschiedene Auswerteverfahren angewendet werden. Dabei kann auch die Qualität und Effizienz der Auswerteverfahren geprüft und optimiert werden.

2.7 Auswertung und Diskussion der Ergebnisse

Die aufgezeichneten Rohdaten bestehen aus den Driftzeiten der Drähte, die bei einem Ereignis einen Impuls gesehen haben. Sie wurden zur Bestimmung folgender Größen ausgewertet:

- Driftzeitspektrum
- Winkelverteilung
- Zeitauflösung
- Ortsauflösung
- räumliche Bahnhäufigkeit

Die Driftzeitspektren für sieben nebeneinanderliegende Drähte sind in Abbildung 2.9 zu sehen. Die unterschiedliche Verteilung ergibt sich aus den geometrischen Gegebenheiten. Damit bei der gegebenen Quellenlage eine Elektronenbahn mindestens 4 Driftzellen überstreicht, muß ein bestimmter, kleiner Winkelbereich gegeben sein. Für diesen Winkelbereich ist für die am Rand liegenden Drähte ein höherer Schwerpunkt im Driftzeitspektrum zu erwarten als bei der mittleren Drähten.

Um den Winkelbereich zu bestimmen, wurde die Häufigkeit der Elektronenbahnen über ihren Winkel bestimmt. Das Ergebnis ist durch die geometrischen Verhältnisse bestimmt. Der mittlere Winkel beträgt etwa 43 Grad bei einer mittleren Breite von 6 Grad.

Die Zeitauflösung ist in Abbildung 2.10 dargestellt. Sie wurde berechnet, indem für jedes auswertbare Ereignis die Zeitdifferenz für je zwei symmetrisch um den Durchstoßpunkt liegende Driftzeitendifferenzen bestimmt und in eine Häufigkeitsverteilung einsortiert wurde. Sie liefert Hinweise auf den Einfluß der Inhomogenität des Feldes in Drahtnähe und die Genauigkeit der Driftzeitbestimmung. Die Halbwertsbreite (FWHM) dieser Verteilung beträgt 34 ns. Daraus ergibt sich eine Zeitauflösung je Draht von 7 ns.

In diesen Wert geht in wesentlichem Umfang die Kleinwinkelstreuung der Elektronen am Gas ein. Die geringe Energie der Elektronen bis max. 2.2 MeV führt dabei zu einem merklichen Effekt. Der Streuwinkel läßt sich für Elektronen abschätzen durch [19]:

$$\theta_0(x, E) = \frac{13.6 \text{ MeV}}{E\beta} \cdot \sqrt{\frac{x}{X_0}} \cdot (1 + 0.038 \ln(\frac{x}{X_0})) \quad (2.6)$$

mit dem Impuls p , der Geschwindigkeit β (in Einheiten von c) und der Ladung e . X_0 ist die Strahlungslänge, die im Medium zurückgelegte Distanz ist x . Für das verwendete Argon-Isobutan-Gemisch beträgt die Strahlungslänge etwa 160 m. Da die Streuung

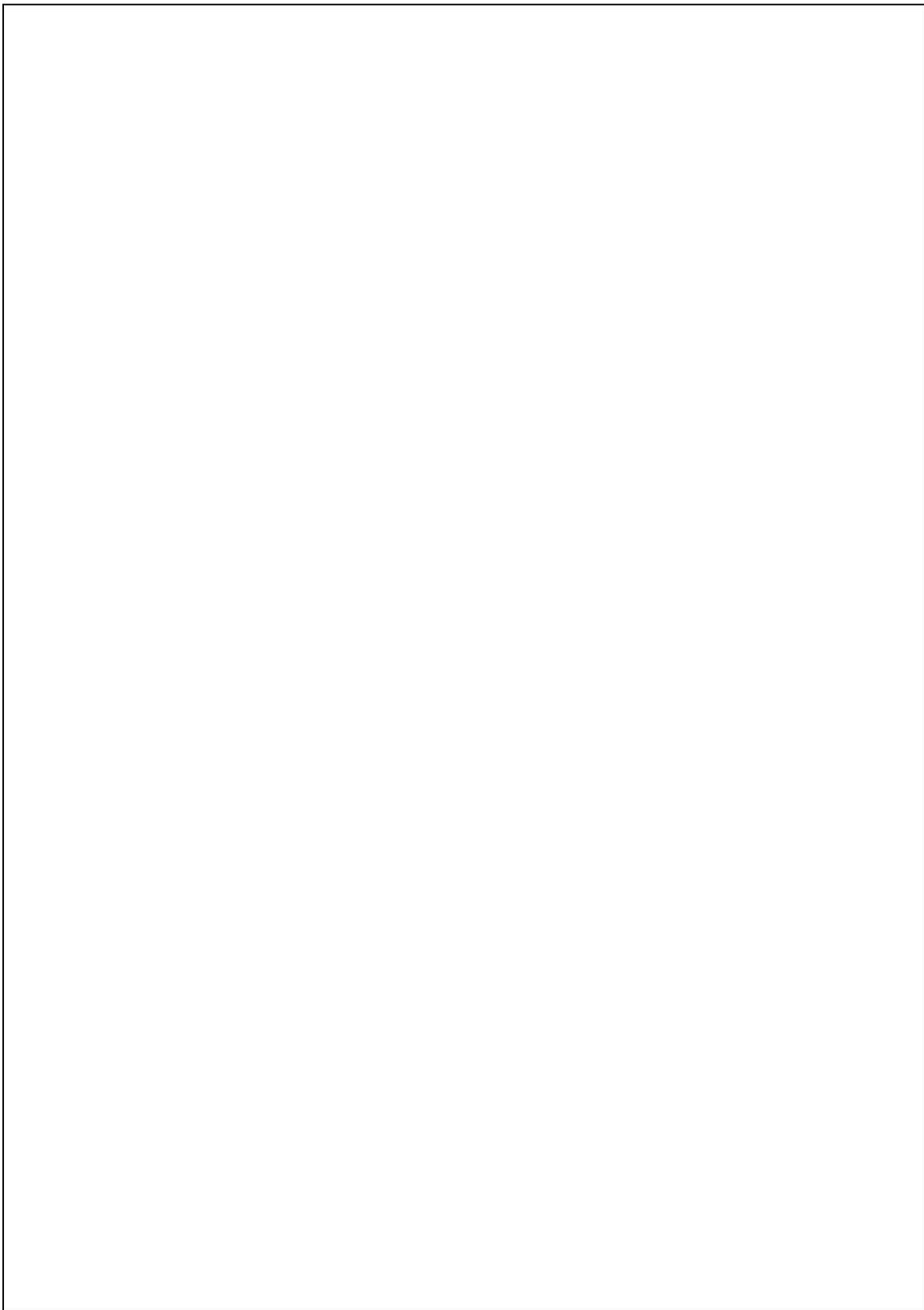


Abb. 2.9: Driftzeitspektren für sieben nebeneinanderliegende Drähte. Zu erkennen ist, daß die Spektren der außenliegenden Drähte einen Schwerpunkt bei längeren Driftzeiten haben.

statistisch verläuft, beträgt die Ablage von der ursprünglichen Bahn $y = 1/\sqrt{3}x\theta_0$. Diagonal fliegende Elektronen laufen etwa 2.8 cm durch die Kammer. Für eine Energie von 1.1 MeV beträgt die Winkelabweichung 120 mrad. Daraus ergibt sich für die durchflogene Distanz eine Bahnabweichung von 2 mm, in Driftrichtung sind das 1.4 mm. Bei einer inversen Driftgeschwindigkeit von 20 ns/mm ergibt sich daraus für die Zeitabweichung ein Wert von 28 ns. Dieser Wert ist zu vergleichen mit der gemessenen Gesamtzeitverteilung von 34 ns. Hinzu kommen noch Fehler durch die Meßelektronik (etwa 1 ns), die Gasdiffusion und die Statistik der Sekundärelektronenerzeugung (etwa 1 bis 2 ns).

Die Größe der Streuung im relativistischen Bereich ist umgekehrt proportional zur Energie der Elektronen. Daher führt der Effekt der Kleinwinkelstreuung für die typische Energie bei Beschleunigerexperimenten zu keiner Einschränkung der Meßgenauigkeit.

Die Steigungsbestimmung wird zusätzlich noch durch Streuung der Elektronen an den der Quelle zugewandten Gasabschluß- und Hochspannungsfolie mit einem Fehler beaufschlagt. Die Foliendicke beträgt insgesamt $8 \mu\text{m}$, die Strahlungslänge in Kunststoff



Abb. 2.10: Driftzeitauflösung. Die an die Meßdaten angepasste Gauß-Verteilung ist mit eingezeichnet.

etwa 40 cm. Der mittlere bei 1.1 MeV durch Kleinwinkelstreuung verursachte Streuwinkel ergibt sich dann nach obiger Gleichung zu etwa 40 mrad. Auch hier gilt wieder, daß der Effekt bei höherer Energie proportional kleiner wird.

Aus der Zeitaufösung von 7 ns resultiert eine Ortsauflösung von 0.4 mm bei einer Driftgeschwindigkeit von 0.05 mm/ns. Da der Durchstoßpunkt auf der Hälfte des Weges des Elektrons durch die Kammer liegt und aus mehreren Driftzeiten bestimmt wird, besitzt er einen Fehler von nur etwa 0.2 mm.

Um die Lage der Quelle zu bestimmen, wurde die räumliche Häufigkeit der Elektronenbahnen in der Kammerebene berechnet. Dazu wurden die Elektronenbahnen nach den Verfahren der linearen Regression und des MIT-Algorithmus ausgewertet. Aus dem sich daraus ergebenden Durchstoßpunkt sowie dem unter Annahme einer konstanten Driftgeschwindigkeit berechneten Steigungswinkel können die sich ergebenden Geraden

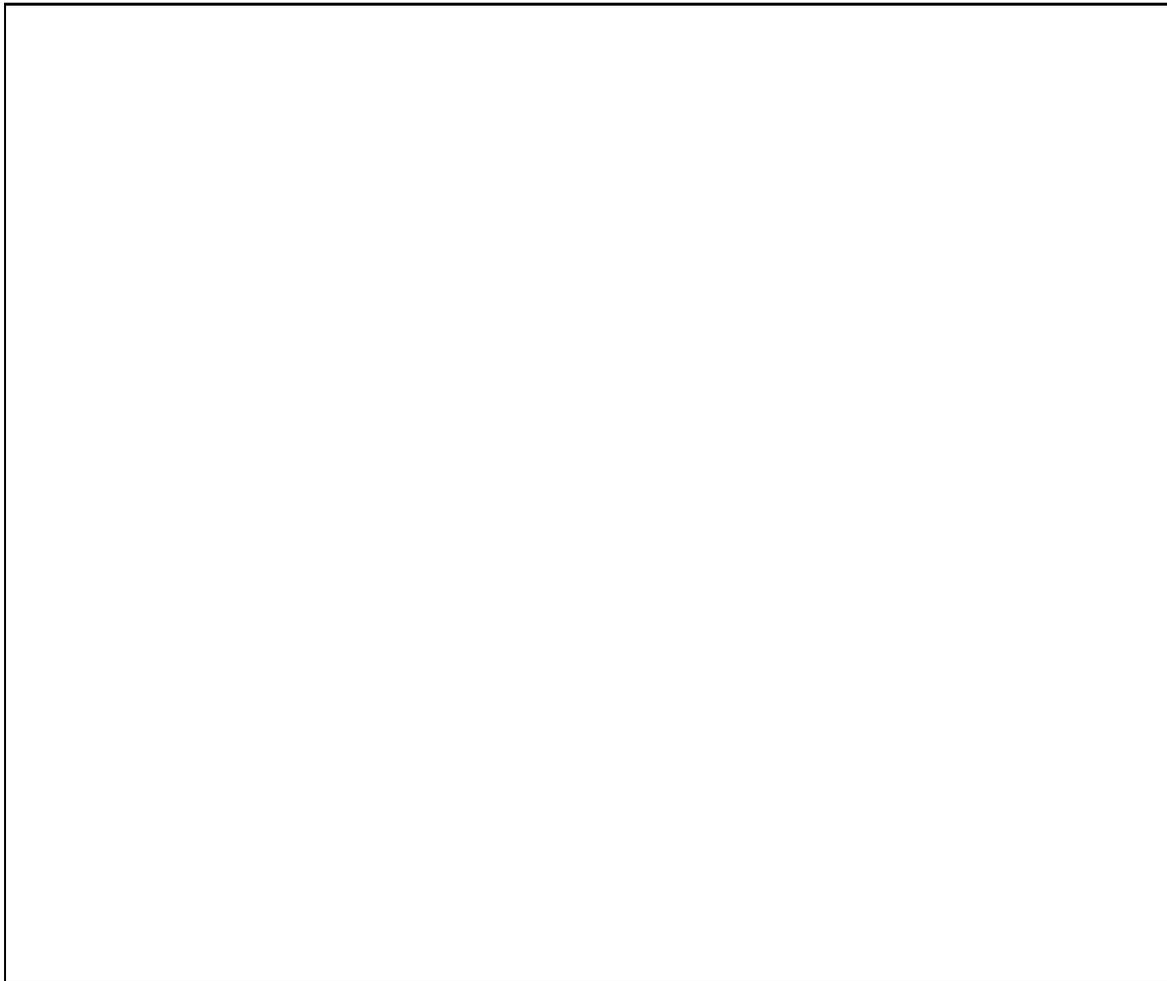


Abb. 2.11: Bahnhäufigkeitsdarstellung. Dunklere Grauwerte entsprechen einer häufiger auftretenden Bahn. Mit eingezeichnet ist die Drahtebene bei $y=0$ mm, die Quellenebene bei $y=-32$ mm (gestrichelt) und die aktive Zone der Kammer mit den aktiven Drähten. Man erkennt, daß alle nachgewiesenen Bahnen durch die aktive Zone laufen. Unterhalb der Quellenebene wird die Verteilung wieder breiter.

über den Bereich der gemessenen Werte hinaus extrapoliert werden. Es ist zu erwarten, daß die Breite der Bahnverteilung an dem Ort der Quelle ein Minimum aufweist. Dazu wurden die bestimmten Geraden in eine zweidimensionale Matrix hineingezählt. Es ergab sich die in Abbildung 2.11 gezeigte Verteilung. Die engste Stelle ist im Abstand von 32 mm von der Drahtebene zu sehen. Dies entspricht auch dem Ort der Quelle.

Bei dieser Auswertung zeigt sich, daß der Fehler, der durch die Verwendung des MIT-Algorithmus gemacht wird, vernachlässigbar gering ist. Merkliche Unterschiede zur Auswertung mit linearer Regression lassen sich nicht feststellen. Dies ist im Hinblick auf eine Online-Reduzierung der Daten von mehreren Driftzeiten auf einen Durchstoßpunkt ein wesentlicher Vorteil. Der MIT-Algorithmus eignet sich durch seine geringe Komplexität sehr gut für eine Online-Anwendung. Außerdem wurde die gesamte Konsistenz der Meßdaten überprüft und bestätigt.

Abschließend kann man sagen, daß die Vieldrahtdriftkammer bei den Testmessungen die zu erwartenden guten Ergebnisse für die Orts- und Zeitauflösung geliefert hat.

3 Entwicklung eines Graphiksystems für die Beschleunigersteuerung

3.1 Motivation und Zielvorstellungen

Die Beschleunigersteuerung stellt hohe Anforderungen an den Bediener. Um ihn optimal zu unterstützen, ist eine schnell reagierende, einfach erlernbare und ergonomische Bedieneroberfläche notwendig. Eine zeitgemäße Realisierung ist mit Hochleistungsgraphiksystemen möglich. Da kommerzielle Graphiksysteme noch sehr teuer sind, wurde am Institut für Kernphysik im Rahmen einer Studienarbeit [28] eine Graphikkarte entwickelt, die sehr hohe Auflösung, Farbfähigkeit und schnellen Bildaufbau auf einer VME-Doppeleuropakarte vereinigt. Für ein vollständiges System benötigt man noch eine CPU- und eine Netzwerkkarte. Eine schematische Darstellung des Graphiksystems ist in Bild 3.1 gezeigt.

Zur Visualisierung der Prozeßzustände gibt es mehrere verschiedene Lösungswege. Ein international standardisiertes Graphiksystem wurde mit dem Graphischen Kern System GKS [29] geschaffen. Hier wurde versucht, die Probleme herstellerabhängiger Graphiksoftware, wie mangelnde Übertragbarkeit auf andere Systeme, uneinheitliche Programmierung, etc. zu umgehen. Mit GKS ist es möglich, ohne großen Aufwand ein nach den Regeln des Standards geschriebenes Programm auf andere Geräte anzupassen. Die Graphikausgabe geschieht in GKS mit Hilfe standardisierter Primitive, die jeweils mit einem bestimmten Satz von Attributen in ihrer Erscheinung verändert werden können. Um vom Ausgabeformat unabhängig zu werden, wird jede Darstellung in GKS zwei Transformationen unterworfen. Die Darstellung in Weltkoordinaten, das sind vom Benutzer gewählte, dem Problem angepasste Koordinaten, wird in ein normiertes Gerätekoordinatensystem projiziert, dessen Werte zwischen 0 und 1 liegen.

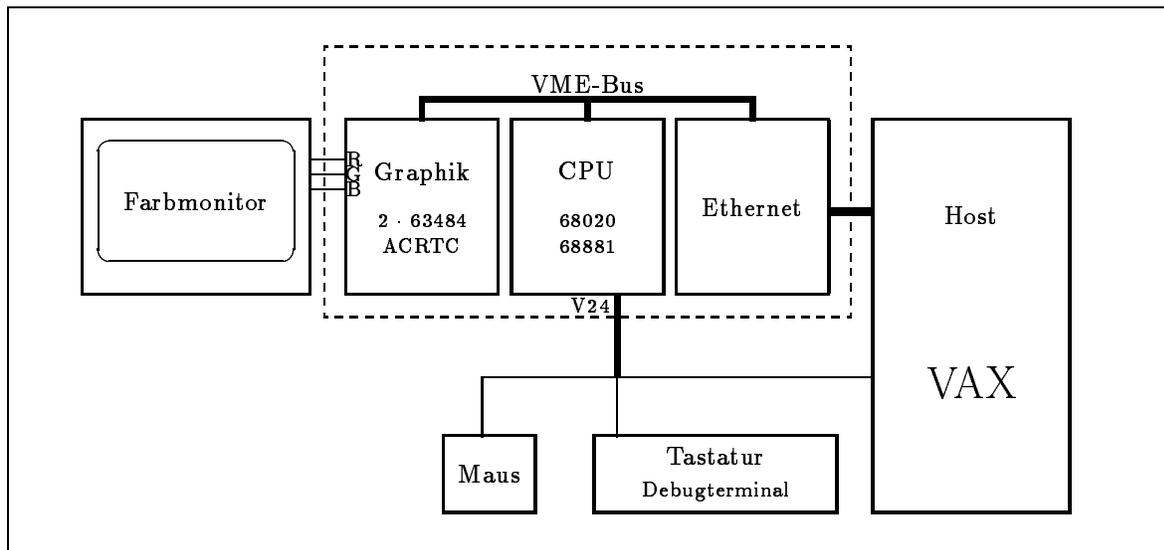


Abb. 3.1: Schema des Graphiksystems

Diese normalisierten Koordinaten werden dann mit der Gerätetransformation auf den jeweiligen Ausschnitt auf dem Darstellungsgerät, also z.B. einem Bildschirm oder Plotter, umgerechnet. Durch diese doppelte Transformation erhält man eine große Unabhängigkeit von den Geräteeinzelheiten. Der GKS-Standard bietet sich durch seine detaillierten Darstellungsmöglichkeiten für komplexe graphische Darstellungen an.

In den letzten Jahren hat sich die intuitive Benutzerführung mit Maus und Bildelemente in Fenstern auf dem Bildschirm sehr verbreitet. Ursprünglich Anfang der 80er Jahre bei Xerox entwickelt, entstanden in den letzten Jahren immer mehr Systeme auf dieser Basis. Im Bereich der vernetzten Workstations hat sich in jüngster Zeit das X Window System [30] als ein de facto-Standard herauskristallisiert. Es basiert auf einem Client/Server-Modell und ermöglicht die verteilte Bearbeitung von Anwendung und Darstellung. Das bedeutet, daß auf einer Workstation mehrere Fenster dargestellt werden, deren Inhalt von Anwendungen auf demselben oder einem entfernten Rechner kontrolliert wird. Das X Window System hat nicht die ausgefeilten graphischen Darstellungsmöglichkeiten wie GKS, ist aber auf die Anforderungen der Windowumgebung hin optimiert. In einem Schichtenmodell liegt es zwischen der Hardware und dem mehr anwendungsorientierten GKS.

3.2 Hardwarebeschreibung

3.2.1 Graphikkarte

Die Graphikkarte wurde im Rahmen einer Studienarbeit am IKDA entwickelt [28]. Die wesentlichen Eckdaten:

- **Zwei Graphikprozessoren:** Durch Verwendung zweier Graphikprozessoren vom Typ Hitachi HD 63484 wird eine hohe Zeichengeschwindigkeit, Adaptierbarkeit an eine Vielzahl verschiedener Monitore durch programmierbare Synchronisation, sowie Entlastung der CPU-Karte von Graphikoperationen erreicht. Die Trennung der zwei Graphikprozessoren in eigenständige Untereinheiten ermöglicht die Darstellung von zwei unabhängigen Ebenen, z.B. einer Text- und einer Graphikebene.
- **1 bis 4 MB Bildspeicher:** Der große Bildspeicher ermöglicht die Darstellung von bis zu $2048 \cdot 2048$ Bildpunkten bei einer Farbtiefe von 8 Bit. Die hier verwendete Graphikkarte ist mit 2 MB Bildspeicher ausgestattet, ist aber bei Bedarf ohne Probleme erweiterbar. Durch einen derart großen Bildspeicher können auch auf Großmonitoren sehr detaillierte Darstellungen ausgegeben werden. Im unsichtbaren Teil des Bildspeichers können Bildelemente, z.B. Zeichensätze oder Hintergrundmuster, zwischengespeichert werden. Der Bildspeicher ist als Dual-Port-RAM ausgelegt. Das bedeutet, daß die Speicherbausteine sowohl vom Graphikprozessor beschrieben werden als auch gleichzeitig zur Darstellung des Bildes ausgelesen werden können. Ein direkter Zugriff durch die CPU ist nicht möglich.
- **Farbtabelle:** Die programmierbare Farbtabelle setzt den 8 bit-Wert eines Bildpunkts in eine analoge Farbe um, die sich aus den drei Grundfarben Rot, Grün,

und Blau zusammensetzt. Dabei sind für jede Farbe 16 bzw. 256 verschiedene Helligkeitsstufen einstellbar. Daraus ergibt sich die Darstellbarkeit von bis zu 4096 bzw. über 16 Millionen Farbtönen. Die Farbtabelle unterstützt zudem noch Blinkeffekte und das Ausblenden bestimmter Bitebenen.

- **VME-Bus Anschluß:** Die Graphikkarte ist an den VME-Bus anschließbar. Dadurch ergibt sich ein modularer Aufbau des Systems. Der Datenaustausch mit der CPU-Karte geschieht durch Registerzugriffe oder mittels des auf der Karte vorhandenen DMA-Bausteins.

Die maximale Bildpunktfrequenz beträgt 128 MHz. Viele logische Elemente sind in PAL-Bausteinen (PAL: Programmable Array Logic) integriert. Dadurch ergibt sich ein verringerter Platzbedarf im Vergleich zu diskreter TTL-Logik und eine erleichterte Anpassung an geänderte Spezifikationen.

3.2.2 CPU-Karte

Die Aufgabe der CPU-Karte besteht in der Verwaltung der ein- und ausgehenden Graphikdaten. Die CPU-Karte wurde im Rahmen einer Diplomarbeit [8] entwickelt und besitzt folgende Daten:

- **CPU Motorola MC 68020:** Dieser Prozessor besitzt 32 Bit Daten- und Adreßbusse. Die Taktfrequenzen reichen von 8 bis 25 Mhz. Auf der hier verwendeten Karte wurde eine 12.5 MHz-Version verwendet. Zur Beschleunigung von Fließkommaoperationen kann optional eine Gleitkommaeinheit MC 68881 verwendet werden.
- **Speicher:** Der Speicherausbau auf der CPU beträgt 256 KB EPROM, 128 KB statisches RAM und 1 MB dynamisches RAM. Bei Bedarf kann auch 4 MB dynamisches RAM durch Verwendung von 1 MB-Modulen bereitgestellt werden. Der statische Speicher ist durch eine Batterie vor Datenverlust bei Stromausfällen gesichert.
- **Betriebssystem pSOS:** Der Betriebssystemkernel pSOS wird zur grundlegenden Prozeßorganisation und zur Verwaltung der Ein- und Ausgabe verwendet.
- **3 serielle Schnittstellen:** Die Karte besitzt 3 serielle asynchrone Schnittstellen. Zwei davon sind mit einem DUART MC 68681, die dritte durch einen MFP MC 68901 realisiert. Die Schnittstellen werden bei dem Graphiksystem zum Anschluß von Tastatur und Maus sowie zur Verbindung mit einem Hostrechner bzw. Terminalserver bei der Terminalemulation verwendet.
- **8 bit Universalport:** Der interruptfähige 8 Bit-Ein-/Ausgang des MFP ist an den VME-Stecker J2 geführt und kann zur Ansteuerung spezieller Geräte verwendet werden. So wurde beispielsweise eine zweikanalige Schrittmotorsteuerung unter Verwendung dieses Ports entwickelt. Auch die im Graphiksystem verwendete Ethernetkarte benutzt diesen Port.

3.2.3 Ethernetkarte

Die Ethernetkarte [8] stellt einen Anschluß an das hochschulweite LAN (Local Area Network) bereit. Dazu besitzt sie einen speziellen Ethernet-Baustein, der das Ethernet-Protokoll CSMA/CD (CSMA: carrier sense multiple access, CD: collision detect) hardwaremäßig implementiert, sowie einen Manchester-Encoder, der die Daten so umordnet, daß auf einer einzigen Leitung sowohl Datenübertragung als auch Synchronisation möglich ist. Des weiteren ist noch ein kleines RAM vorhanden, das als Zwischenspeicher dient. Der Datenaustausch mit der CPU geschieht über einen spezialisierten Datenbus mit 16 Bit Breite und 20 Bit Adressen. Dadurch wird der VME-Bus nicht belastet.

3.2.4 Tastatur und Maus

Um die Eingabemöglichkeiten an die emulierten Geräte anzugleichen, wurde eine DEC LK201-Tastatur an das Graphiksystem angeschlossen. Diese Tastatur besitzt einen eigenen Mikroprozessor, der über eine serielle Schnittstelle mit 4800 Bd mit der Außenwelt kommuniziert. Es können durch Senden von Kommandos an die Tastatur Modusveränderungen vorgenommen werden. Die beim Drücken von Tasten gesendeten Zeichen entsprechen nicht der standardisierten ASCII-Kodierung. Sie müssen daher erst umkodiert und an eventuell aktivierten Umschaltungen angepasst werden. Die Stromversorgung wird mit 12 V über das Tastaturkabel sichergestellt.

Als Zeiger-Eingabegerät ist eine Standardmaus mit 3 Tasten angeschlossen. Sie sendet bei Bewegung bzw. bei Drücken und Loslassen von Tasten ein Datentelegramm von 5 Bytes mit 1200 Bd. In dem Telegramm ist der Zustand der Tasten und die relative Bewegung sowie die Geschwindigkeit enthalten. Eine gesonderte Stromversorgung ist nicht notwendig.

3.2.5 Monitor

Der Monitor besitzt eine Bandbreite von 100 MHz. Daraus ergibt sich eine darstellbare Zahl von $1280 \cdot 1024$ Bildpunkten bei einer Bildfrequenz von 60 Hz. Die Konvergenz wird durch eine digitale Konvergenzeinheit sichergestellt. Der Anschluß der Bildsignale wird über drei Koaxialleitungen mit 75Ω Impedanz für die drei Grundfarben Rot, Grün und Blau vorgenommen. Dabei werden dem Grün-Signal die Impulse für die Bildsynchronisation überlagert.

3.2.6 Aufbau und Änderungen

Zur Anpassung an die gewählte Konfiguration waren einige Schaltungserweiterungen notwendig. Ein VME-Bus-Arbiter wurde auf die CPU-Karte integriert. Dieser Baustein wurde durch ein PAL realisiert. Er ist im VME-Bussystem notwendig, um die Buszuweisung für verschiedene Busmaster zu überwachen. An den Schnittstellenstecker wurde eine abgesicherte Versorgungsspannung von 12 V für die Tastatur gelegt. Der Stromverbrauch der Maus wird aus den Schnittstellensignalen gedeckt. Zum Anschluß aller Einheiten wurde ein spezieller Schnittstellenverteiler gebaut.

3.3 Grundlagen zur Software

In diesem Abschnitt werden die relevanten Grundbegriffe für die einzelnen Programmodule eingeführt und näher erläutert.

3.3.1 Terminalemulation

Eine Terminalemulation stellt Funktionen zur Verfügung, die gebräuchliche Programme verwenden. Da es eine Vielzahl in Hardware und Funktionsumfang unterschiedlicher Terminals gibt, existieren auch diverse verschiedene Befehlsgruppen. Herstellerspezifische Standards stehen solchen, die herstellerunabhängig genormt wurden, gegenüber. Die am IKDA meistverwendeten Terminals sind aus der DEC VT-Serie (z.B. VT 100, VT 220, VT 320) bzw. dazu kompatibel. Die Textfunktionen des VT-Terminaltyps VT 100 sind in einer ANSI-Norm festgehalten. Zusätzlich dazu sind noch einige Graphikerweiterungen für VT 100-Terminals von der Firma Retrographics vorhanden, die auch mit der Bezeichnung VT 640 benannt werden. Auf dieser Terminalfamilie laufen alle Anwendungen von VMS, z.B. Texteditoren, etc.. Auch spezielle Graphikprogramme, die zur Auswertung kernphysikalischer Messungen verwendet werden, setzen solche Terminals voraus.

Um das Graphiksystem effizient in die vorhandene Umgebung zu integrieren, ist es daher sinnvoll, die Funktionen und Steuersequenzen, die diese Terminals verwenden, zu emulieren, d.h. nachzubilden. Dabei müssen als Grundlage alle Fähigkeiten eines zeichenorientierten Terminals implementiert werden. Darauf bauen die speziellen Steuersequenzen des zu emulierenden Terminals auf, die besondere Funktionen auslösen. Zusätzlich sind die sonstige Funktion nicht beeinträchtigende, d.h. kompatible Erweiterungen vorzusehen, die die speziellen Fähigkeiten der Hardware, wie Zooming, Farbe, komplexe Graphikelemente, etc. zugänglich machen.

Bei der Implementierung jeder Funktion muß abgewogen werden, ob der Aufwand in einem vernünftigen Verhältnis zum Ergebnis steht. Von üblichen Programmen nicht benötigte Funktionen, die einen sehr hohen Aufwand erfordern würden, können im Zweifelsfall wegfallen oder vereinfacht werden. Viel verwendete Funktionen hingegen müssen nicht nur völlig funktionskompatibel sein, sondern sind auch in ihrem Geschwindigkeitsverhalten zu optimieren.

3.3.2 GKS

GKS ist ein ISO- und DIN-Standard (DIN/ISO 7942 [29]). Die Entwicklung dieses Graphikstandards begann in der Mitte der siebziger Jahre und wurde 1983 abgeschlossen. Damals war der Computergraphikbereich von herstellerabhängigen Lösungen dominiert. Die Portierung einer Anwendung auf ein anderes System war nur mit hohem Änderungsaufwand möglich. Daher entschloß man sich, einen offenen Standard zu definieren, der alle Gerätetypen berücksichtigt, spezielle Eigenschaften unterstützt und auf möglichst vielen Systemen implementiert werden kann. Die Hauptbereiche, mit denen sich der GKS-Standard beschäftigt, sind:

- **Ausgabe:** Eine wesentliche Aufgabe eines Graphiksystems ist die Darstellung von Bildern. Dies geschieht in GKS durch Ausgabefunktionen. Die ein Bild aufbauenden Objekte sind Ausgabeprimitive, deren Aussehen (z.B. Linienbreite, Farbe) auf der Darstellungsebene durch zugehörige Attributsätze festgelegt wird.

- **Koordinatensysteme und Transformationen:** Die Primitive werden in einem oder mehreren benutzerdefinierten Koordinatensystemen ausgegeben. Diese Ausgaben müssen bei verschiedenen Geräten in verschiedene Gerätekoordinaten transformiert werden. Der Weg der graphischen Ausgabe vom Anwendungsprogramm zur Darstellungsfläche wird durch zwei Transformationen geräteunabhängig. Von den benutzerspezifischen Weltkoordinaten WC gelangt man durch die Welttransformation zu normalisierten Gerätekoordinaten NDC. Diese werden durch die Gerätetransformation in die Gerätekoordinaten DC überführt.

- **Workstations:** Jedes Ausgabe- und/oder Eingabegerät wird als eine Workstation angesehen, deren Fähigkeiten in einer Beschreibungstabelle festgehalten sind. Deren Inhalt wird dem Anwendungsprogramm durch Abfragefunktionen mitgeteilt. Beispiele einer Workstation sind ein Graphikterminal, ein Plotter, oder ein Graphiktablett.

- **Eingabe:** Es werden sechs Eingabearten mit je drei Eingabeklassen unterschieden, die jedem Bedarf gerecht werden sollten. Für jeden Eingabeart gibt es mehrere Arten der Echo-Darstellung. Damit können sowohl alphanumerische, graphische, als auch auswählende Daten mit einer Vielzahl unterschiedlicher Eingabegeräte erzeugt werden.

- **Segmentierung:** Zur weiteren Bearbeitung von Bildteilen verwendet man Segmente. Diese können einzeln transformiert, kopiert, in ihren Attributen verändert und gelöscht werden. Zur zentralen Speicherung von Segmenten existiert ein Workstation-unabhängiger Segmentespeicher (WISS).

- **Metadateien:** Um Bilder auf Dauer festzuhalten und auf andere Systeme zu übertragen, können die Bildelemente in eine Datei geschrieben werden. Der Aufbau dieser Datei ist in einer weiteren Norm festgelegt.

- **Funktionsstufen:** Für Ein- und Ausgabe sind jeweils 3, in ihrem Umfang aufsteigende Funktionsstufen möglich, Dies ergibt zusammen 9 verschiedene Stufen, die je nach Anwendung zu wählen sind. Die Ausgabestufen werden mit Ziffern von 0 bis 2, die Eingabestufen mit Buchstaben von a bis c bezeichnet. Die Stufe mit der höchsten Funktionalität ist 2c.

- **Erweiterbarkeit:** Um zukünftige Entwicklungen wie z.B. 3D-Erweiterungen nicht auszuschliessen, gibt es die Möglichkeit, spezielle Kommandos zu verwenden, die dann allerdings nicht genormt sind und nicht von allen Implementationen unterstützt werden müssen.

3.3.3 GKS-Applikationsschnittstelle

Die Schnittstelle von GKS zu Anwendungen ist als Funktionsbibliothek aufgebaut, deren exakte Syntax sprachunabhängig genormt ist [29]. Die Implementation in einer speziellen Programmiersprache muß auf deren Besonderheiten Rücksicht nehmen und ist daher in jeder Sprache verschieden. Für die Fortran- und Pascal-Schnittstelle existiert ein Normvorschlag. In FORTRAN werden GKS-Funktionen als Unterprogramme mit den jeweiligen Ein- und Ausgabeparametern aufgerufen. Diese Art der GKS-Implementation hat den Vorteil, daß die Programmierung mit einer bekannten Sprache erfolgen kann und keinen Einschränkungen unterliegt. Die Fortran-Unterprogramme können bei dem hier verwendeten Betriebssystem auch von anderen Programmiersprachen im Rahmen des standardisierten Aufrufs (VMS-calling standard) angesprochen werden.

Die Funktionen gliedern sich in die Gruppen Zustandskontrolle, Transformation, Ausgabe, Eingabe, Attribute, Abfrage und Fehlerbehandlung. Eine genaue Beschreibung aller GKS-Funktionen und der ihnen zugrundeliegenden Philosophie ist zum Beispiel in [31] zu finden. Eine Liste wichtiger Funktionen, nach Gruppen sortiert, ist in Tabelle 3.1 zu finden.

Tabelle 3.1: Wichtige GKS-Funktionen und ihre Fortran-Namen:

Name	GKS-Funktion	Ausgelöster Treiberaufruf
GOPKS	Open GKS	–
GOPWK	Open Workstation	G\$OPEN
GACWK	Activate Workstation	–
GDAWK	Deactivate Workstation	–
GCLWK	Close Workstation	G\$CLOSE
GCLKS	Close GKS	–
GPL	Polyline	G\$LINE
GPM	Polymarker	G\$LINE
GTX	Text	G\$TEXT
GFA	Fill Area	G\$PLOT
GCA	Cell Array	G\$PLOT
GGDP	Generalized Drawing Primitive	G\$CIRCLE

3.3.4 GKS-Gerätetreiberschnittstelle

Die Anwendungsfunktionen werden vom geräteunabhängigen Teil bearbeitet. Dabei ist je nach Funktion auch eine Interaktion mit den Workstations nötig. Diese geschieht über die Gerätetreiberschnittstelle. Sie trennt den geräteunabhängigen Teil (DI=Device Independent) vom geräteabhängigen Teil (DD=Device Dependent) und wird daher auch DI/DD-Schnittstelle genannt. Diese Schnittstelle ist nicht definitiv genormt und

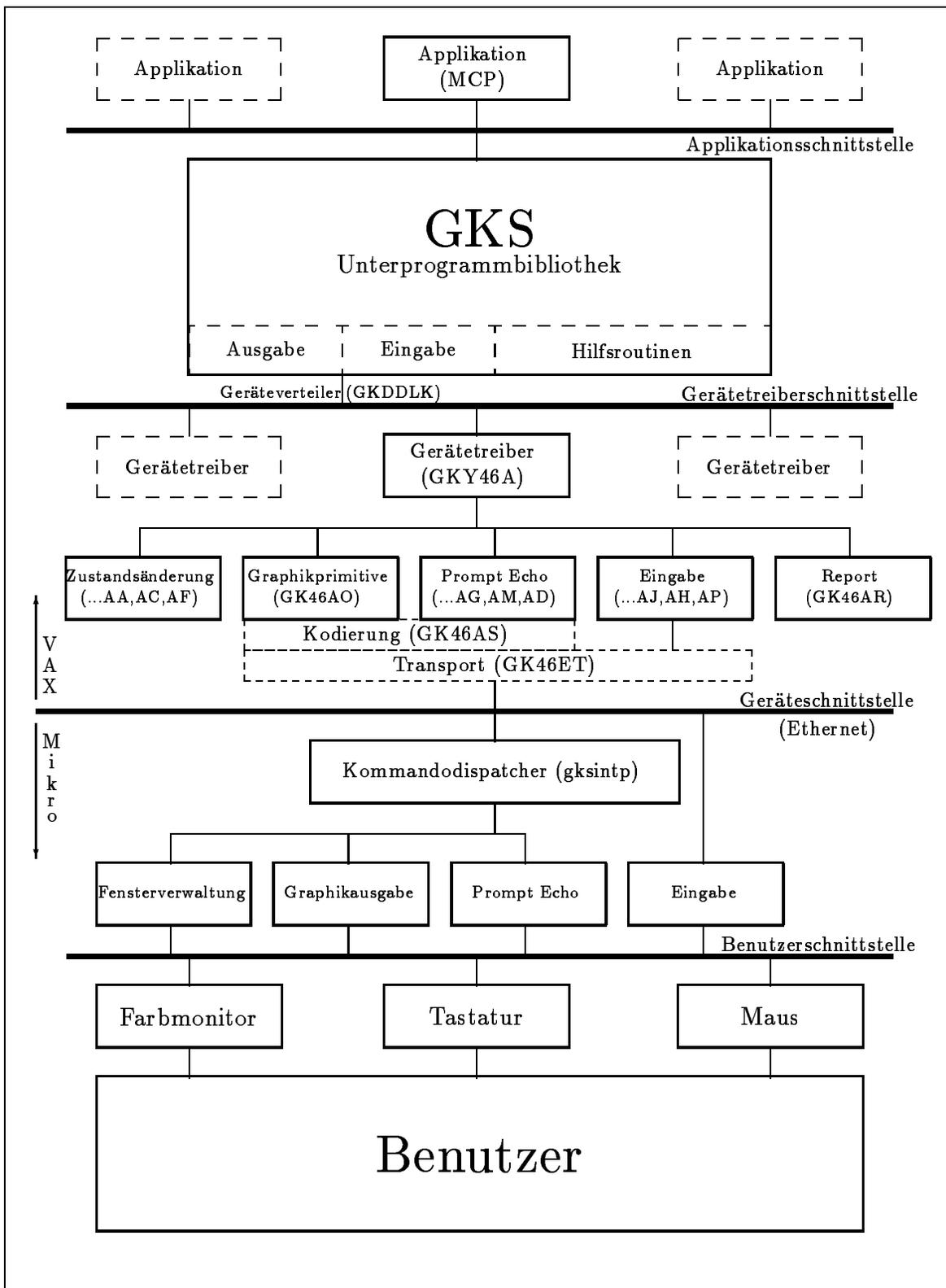


Abb. 3.2: Struktur des GKS-Systems mit Aufteilung in Funktionsebenen, die durch definierte Schnittstellen kommunizieren. Durch Anpassung der geräteabhängigen Teile können neue Geräte eingebunden werden.

daher je nach Implementation verschieden. Die bei der verwendeten Implementierung von GKS vorhandene Gerätetreiberschnittstelle ist in [33] dokumentiert.

An die GKS-Gerätetreiberschnittstelle schließt sich der im Rahmen dieser Diplomarbeit entwickelte Treiber an. Die Funktionen haben prinzipiell Entsprechungen mit denen der Anwendungsschnittstelle. Einige sind jedoch nicht vorhanden, bzw. besitzen andere Argumente. Die verwendeten Koordinaten sind grundsätzlich Gerätekoordinaten. Transformationen geschehen schon in den übergeordneten Schichten. Eine Darstellung der Struktur des GKS-Systems mit den genannten Schnittstellen wird in Abbildung 3.2 gezeigt.

3.3.5 Gerätetreiber

Alle Gerätetreiber sind nach einem festen Gerüst aufgebaut, in das die jeweiligen Besonderheiten des Geräts eingearbeitet werden. Das vom allgemeinen Schnittstellenmodul gerufene gerätespezifische Hauptmodul des Gerätetreiber verzweigt in die nach die Funktionskategorien geordneten Funktionsmodule. Sie rufen bei Bedarf weitere Unter-routinen auf, die mit dem Gerät kommunizieren oder sonstige überschaubare Funktionen ausführen. Durch diese Modularisierung wird der Gerätetreiber übersichtlich strukturiert. Dadurch können neue Geräte relativ einfach von einem Standardtreiber abgeleitet werden. Die Programmiersprache ist Fortran.

3.3.6 Fenstertechnik

Zur Anpassung an das MCP-Programm wurde die Darstellung mehrerer übereinanderliegender Bildauschnitte (Fenster) notwendig. Jedes Fenster entspricht einer virtuellen Workstation, die von GKS getrennt angesprochen und verwaltet wird.

Die Fensterverwaltung muß die Ausgaben und Eingaben den jeweiligen Fenstern zuordnen. Dabei müssen die jeweiligen Fenstergrenzen berücksichtigt werden. Sichtbare und unsichtbare Regionen sind bei Ausgabe zu unterscheiden. Verdeckte Regionen müssen zwischengespeichert werden, damit bei einem Aufdecken der Fensterinhalt unbeschädigt zurückkopiert werden kann. Um in eine verdeckte Region eines Fensters schreiben zu können, muß die Möglichkeit bestehen, in den Zwischenspeicher zu zeichnen. Wenn dies nicht möglich ist, muß das zu ändernde Fenster immer erst vollständig sichtbar gemacht werden.

3.3.7 Kommunikation

Die Kommunikation zwischen den Modulen auf den verschiedenen Rechnern erfordert eine besondere Betrachtung, um die gesteckten Geschwindigkeitsanforderungen zu erfüllen.

Um einen schnellen Bildaufbau zu erreichen, muß die Schnittstelle zwischen VAX und Mikro eine hohe Bandbreite aufweisen. Die übliche serielle V24-Leitung mit einer Datenrate von 19200 Baud ist dabei meist nicht mehr ausreichend. Daher wurde in das Graphiksystem eine Ethernetkarte integriert. Ethernet besitzt eine Rohdatenrate

von 10 MBaud. Die Nettodatenrate beträgt je nach Netzbelastung und Protokoll nur etwa ein Zehntel dieses Wertes. Sie ist aber trotzdem noch wesentlich höher als bei der V24-Kopplung.

Im Gegensatz zu seriellen Verbindungen, bei denen jedes Terminal eine eigene Verbindung zu einem Schnittstellenmultiplexer am Rechner besitzt, ist Ethernet eine einzige Leitung, an der alle Teilnehmer angeschlossen sind. Jeder Teilnehmer ist durch eine eindeutige, 6 Byte lange Adresse identifiziert. Die Daten werden in größeren Einheiten (max. 1500 Bytes) ausgetauscht, an deren Anfang jeweils die Ziel- und Quelladresse steht. Solange eine Station das Netz verwendet, darf keine andere Station senden. Falls zufällig mehrere Stationen gleichzeitig zu senden anfangen, entsteht eine Kollision, die von den sendenden Stationen bemerkt wird. Dann wartet jede Station eine gewisse Zeit, deren Länge durch einen Zufallsgenerator bestimmt wird, und versucht dann die Übertragung erneut. Die Länge eines Ethernetsegments ist auf maximal 1500 m begrenzt, um die Signallaufzeiten nicht zu lang werden zu lassen. Maximal können 500 Stationen pro Ethernetsegment angeschlossen werden. Mehrere Ethernetsegmente können bei Bedarf mit speziellen Brückenbausteinen, die Datenpakete von einem Segment in ein anderes weiterleiten, verbunden werden.

3.4 Entwicklung der Systemsoftware

Die Programmteile wurden auf einer VAX unter VMS in der Programmiersprache C sowie in 68020-Assembler erstellt. Dabei kam ein Cross-Development-System der Firma Microtek zum Einsatz. Die gelinkten und mit Teilen der Betriebssystemumgebung versehenen Programme liegen als S-Records (spezielle Kodierungsform von binären Daten in lesbaren ASCII-Zeichen mit Anfangsadresse und Prüfsumme) vor, die über serielle Kopplung oder Ethernet in das Graphiksystem geladen werden. Zu diesem Zweck und zum Austesten der Programme besitzt das System den Debugger pROBE. Dieser ist Teil der pSOS-Systemsoftware. Als weiteres Teil ist der pSOS Multitasking-Realtime-Kernel vorhanden, der Systemfunktionen bereitstellt. Beides liegt im ROM des Graphiksystem. Die Programme werden mit dem Debugger gestartet.

Die Entwicklung verlief in Stufen mit wachsender Funktionalität. Zuerst wurden die hardwarenahen Register zur Speicherorganisation und Bildsynchronisation der Graphikkarte gesetzt und das Zusammenspiel der beiden Graphikprozessoren, von denen einer den anderen synchronisiert (Master/Slave), optimiert. Die Implementation der Textfonts folgte. Aufbauend auf diese Low-Level-Funktionen wurde ein Konsolenterminal ohne besondere Funktionen entwickelt. Dies wurde dann um die Kontrollsequenzen und Terminalemulationen erweitert. Der Interpreter für die GKS-Funktionen wurde getrennt entwickelt und dann in den bestehenden Terminalemulator integriert. Parallel dazu entstand die VAX-Seite des GKS-Treibers, die eng mit dem Terminalemulator zusammenarbeitet. Zum Schluß wurde die Ethernet-Kopplung implementiert, nachdem zuvor alle Steuersequenzen über die serielle Leitung ausgetestet waren.

3.5 Integration in die Applikationsumgebung

Hauptanwendung für das Graphiksystem ist die Beschleunigersteuerung. Daneben ist die Auswertung von Experimenten sowie die Bedienung des Spektrometers als Anwendung denkbar. Das Beschleunigersteuerprogramm MCP sowie das Meßsystem GOOSY verwendet zur graphischen Darstellung die GKS-Implementation von GTSGRAL. Daher wurde ein Treiber für diese Implementation geschrieben (nachfolgend ACRTC-Treiber genannt). Das Beschleunigersteuerprogramm ist für die Verwendung auf einer DEC VWS-Workstation (DEGPX) geschrieben. Dazu wurden speziellen Eigenschaften des Treibers für die VWS-Workstation (nachfolgend DEGPX-Treiber genannt) benötigt, zu denen unter anderem folgendes zählt:

- **Windows als GKS-Workstations:** Die Verwendung mehrerer Windows wurde aus GKS-Sicht als eine Vervielfältung des Workstationtyps gelöst. In einer Datei werden die Position und die Größe des Windows festgelegt. Die einzelnen Windows werden dann als unabhängige GKS-Workstations behandelt.
- **Spezieller Echomodus für Pulldown-Menüs:** Um die Verwendung von Pulldown-Menüs optisch ansprechend zu gestalten, wurde ein spezieller Echomodus für Choice-Eingaben in den DEGPX-Treiber integriert, der die Echoregion zwischenspeichert und eine gefülltes Rechteck unter die Echoregion legt.
- **Spezieller Prompt-Echo-Typ für Locator-Eingabe:** Zur Auswahl von Zoom-Windows in der Beschleunigerdarstellung war es notwendig, rechteckige Bereiche auszuwählen. Dazu wurde dem DEGPX-Treiber der Prompt-Echo-Typ Rubber-Band-Rectangle nachträglich hinzugefügt.

Diese Besonderheiten mußten auch vom ACRTC-Treiber unterstützt werden. Als besonders aufwendig erwies sich die Windowtechnik, da der DEGPX-Treiber hier voll auf dem UIS-Windowssystem aufbaut und sich daher um die Verwaltung der Fenster nicht kümmern muß. Dazu gehören das Zwischenspeichern verdeckter Teile, das Widersichtbarmachen aufgedeckter Teile, die Änderung der Window-Priorität, die Zuordnung der Eingaben zu den Windows, und weitere kleinere Aufgaben. Diese Fensterverwaltung mußte daher völlig neu geschrieben werden. Dabei stellte sich heraus, das der Graphikprozessor für solche Aufgaben eigentlich nicht konstruiert ist. Die Fensterverwaltung erfordert nämlich das schnelle Umspeichern großer rechteckiger Bildausschnitte. Bei dieser im Grunde anspruchslosen Operation ist der Graphikprozessor recht langsam, da er den Speicher nur in 16 Bit-Worten mit einer geringen Frequenz zugreifen kann. Seine eigentliche Domäne liegt im schnellen Zeichnen von graphischen Elementen. Der Ausweg über den schnelleren Hauptprozessor ist verbaut, da der Bildspeicher nur für den Graphikprozessor zugänglich ist. Modernere Graphikprozessoren sind auf die schnelle Verarbeitung von Rasteroperationen hin optimiert, da diese durch die rasante Verbreitung der Windowssysteme einen wesentlichen Anteil an der Systemleistung haben.

Um daneben die universelle Verwendbarkeit mit vielen sonstigen Programmen zu erreichen, wurde die Terminalemulation entwickelt. Dabei wurde Wert auf einen schnellen

Bildaufbau und eine hohe Scrollgeschwindigkeit gelegt. Dies war nur durch Verwendung spezieller Bildsegmentregister des ACRTC möglich.

3.6 Funktionsbeschreibung

3.6.1 Terminalemulation

Der Funktionsumfang der Terminalemulation umfaßt den vollständigen Befehlssatz eines VT100-Terminals mit Erweiterungen von VT220 und VT640 bzw. Tektronix 4014. Außerdem sind Erweiterungen zur Unterstützung der spezieller Eigenschaften der Hardware vorhanden. Dazu gehören unter anderem verschiedene Schriftgrößen, erweiterte Graphikkommandos und Zugriff auf die Farbpalette. Die realisierten Escape-Sequenzen werden im Anhang aufgelistet.

Die Tasten liefern dieselben Codes wie bei einem VT 220-Terminal. Die Codes können daher einem VT 220-Referenzhandbuch (z.B. [34]) entnommen werden.

Ein lokaler Setupmodus wurde nicht implementiert. Das Terminal besitzt die optimale Einstellung für den Betrieb im VAXcluster. Die Baudrate der Hostschnittstelle beträgt 19200 Bd. Ein Rücksetzen der Kommunikation ist mit der F3 -Taste unter gleichzeitigem Drücken der Control -Taste möglich. Ein Rücksetzen der Tastatur ist mit der F3 -Taste unter gleichzeitigem Drücken der Shift - und der Control -Taste möglich. Die F5 -Taste löst einen Break aus.

3.6.2 GKS-Treiber

Der Funktionsumfang des GKS-Treibers orientiert sich am Modell des DEGPX-Treibers. Eine Beschreibung der DI/DD-Schnittstelle, der Geräteeigenschaften, und der implementierten GKS-Modi befindet sich im Anhang. Die standardisierte GKS-Gerätetabelle entspricht der des DEGPX-Treibers mit den im Anhang beschriebenen Änderungen.

Die Aufteilung der Funktionen auf den VAX- und den VME-Teil wurde anhand der DI/DD-Schnittstelle vorgenommen. Der Gerätetreiber auf VAX-Seite behandelt alle Request-Funktionen ohne Zugriff auf den Mikro. Alle Ausgabeprimitive werden an den Mikro weitergereicht, ohne weiter behandelt zu werden. Für die Verwaltung von Eingaben werden auf VAX-Seite entsprechende QIOs bereitgestellt. Die Eingaben werden von AST-Routinen in einer FIFO-Queue gepuffert, aus der die übergeordneten GKS-Funktionen bei Bedarf die Daten abholen. Eingaben werden nicht quittiert. Eine Tabelle der im Mikroteil ausgeführten Routinen ist im Anhang zu finden.

3.7 Test

Durch den sequentiellen Ausbau der Funktionalität hatte man immer nur einen überschaubaren Teil der Software auf ihre Zuverlässigkeit hin zu überprüfen. Dies erleichtert die Fehlersuche sehr, da bei Änderung an mehreren verschiedenen Programmteilen schwierige Situationen auftreten können, deren Ursache nicht leicht zu finden ist.

In jeder Entwicklungsphase wurden entsprechende Testprogramme entwickelt, die die Funktionen unter möglichst vielen Anwendungszuständen testen können. Außerdem wurden bestehende Anwendungen wie GCOPY, PLOTDATA, etc. zur Verifikation unter realen Bedingungen verwendet.

Der GKS-Treiber wurde erst mit den bei GKSGRAL mitgelieferten Testprogrammen, die viele Funktionen (z.B. Ausgabeprimitive, Eingabe, Segmentierung) testen, geprüft. Nach erfolgreichem Bestehen dieser Tests wurden dann GKS-Anwendungsprogramme, insbesondere das MCP-Beschleunigerkontrollprogramm, auf ihre volle Funktion hin untersucht.

3.8 Weitere Entwicklungsmöglichkeiten Software

Aufgrund des modularen Aufbaus des Graphiksystems sind Weiterentwicklungen der Hard- und Software mit geringem Aufwand möglich. Beispielsweise kann ein Server für die sich stetig weiterverbreitende graphische Benutzeroberfläche X Window implementiert werden. Die Eigenschaften des Graphikprozessors sind dabei allerdings für eine zügige Implementation eher hinderlich. Ein vom MIT gelieferter X Window-Server [30] ist auf den direkten Zugriff auf den Bildspeicher durch den Hauptprozessor ausgelegt. Dies ist bei der Graphikkarte nicht möglich. Außerdem ist die Betriebssystemanbindung auf UNIX-Systeme ausgelegt. Die Betriebssystemfunktionen des pSOS-Kernels sind nicht direkt kompatibel. Dadurch ist die Komplexität der Anpassung sehr hoch. Auf Dauer wird sich jedoch die X Window-Benutzeroberfläche durchsetzen, wodurch die Attraktivität einer X Server-Implementation sehr hoch wird.

3.9 Weitere Entwicklungsmöglichkeiten Hardware

3.9.1 Speichererweiterung

Die Erweiterung des Speicher der CPU-Karte auf 4 MB ist durch Austausch der 256 KB-Module durch 1 MB-Module möglich. Die Softwareparameter müssen angepasst werden. Es steht dann entsprechend mehr Speicher für das Zwischenspeichern von Hintergrundinhalten zur Verfügung. Für die X Server-Implementation ist diese Erweiterung auch sinnvoll. Der Speicher der Graphikkarte kann auf maximal 4 MB erweitert werden. Dies erlaubt den Anschluß höherauflösender Bildschirme (bis $2048 \cdot 2048$ Bildpunkte) oder die Zwischenspeicherung von ganzen Bildinhalten oder von Textfonts.

3.9.2 Mehrere Bildschirme

Das VME-Buskonzept erlaubt den Anschluß vieler Erweiterungskarten. Durch Umkodierung der Graphikkarte können theoretisch bis zu 256 Graphikkarten von einer CPU-Karte adressiert werden. Dies ermöglicht beispielsweise die Verwendung eines weiteren Bildschirms. Eine fertige, mit 1 MB bestückte Graphikkarte mit einem 19 Zoll-Farbbildschirm mit $1024 \cdot 768$ Bildpunkten ist zur Realisierung dieser Option schon vorhanden. Alternativ ist natürlich auch der Einsatz einer weiteren CPU-Karte

und einer Ethernetkarte möglich, um so ein komplettes, autarkes Graphiksystem aufzubauen.

3.9.3 Farberweiterung

Durch einfachen Austausch der verwendeten Farbpalette durch einen pinkompatiblen Typ ist die Erweiterung der Farbvielfalt von 4096 auf 16777216 mögliche Farbtöne möglich. Für den vorgesehenen Einsatz sollte jedoch der gebotene Farbtonvorrat ausreichen.

4 Status und Schlußbemerkung

Die Vieldrahtdriftkammern sind in das Spektrometer eingebaut und haben ihre Funktionstüchtigkeit auch mit dem Beschleuniger schon in einer Strahlzeit unter Beweis gestellt. Die Auflösungsweite waren dabei aufgrund der höheren Energie des Beschleunigerstrahls von 30 bis 75 MeV erheblich besser, da die Kleinwinkelstreuung bei dieser Energie vernachlässigbar ist.

Das Graphiksystem ist fertig aufgebaut und einsatzbereit. Es ist in das Kontrollpult des neuen Beschleunigers integriert. Zur Zeit wird die Anbindung eines zweiten Bildschirms an den GKS-Treiber implementiert. Eine Implementierung des X Window Systems wurde noch nicht in Angriff genommen.

Anhang

A Dokumentation Terminalemulation

Auf den folgenden Seiten werden die realisierten Steuersequenzen tabellarisch aufgeführt und kurz erklärt.

A.1 ANSI/VT Modus

A.1.1 Modus einschalten

ESC [2h	Tastatur sperren
ESC [4h	Einfügemodus
ESC [?1h	Cursortasten Applikationsmodus
ESC [?3h	132 Spalten
ESC [?8h	Autorepeat ein
ESC [?38h	Tektronix Graphikmodus ein

A.1.2 Modus ausschalten

ESC [2l	Tastatur entsperren
ESC [4l	Überschreibmodus
ESC [?1l	Cursortasten Standardmodus
ESC [?3l	80 Spalten
ESC [?8l	Autorepeat aus
ESC [?38l	Tektronix Graphikmodus aus

A.1.3 Zeichensatz Vorwahl

Das Format ist: ESC *primary final*

Dabei wählt *primary* einen von zwei logischen Zeichensätzen (G0, G1) aus, und *final* wählt den in diesem darzustellenden Zeichenvorrat aus.

Primary Auswahl

(G0
)	G1

Final Auswahl

A	UK national
B	US-ASCII
0	Spezial Graphik

A.1.4 Auswahl Zeichensatz zur Darstellung

SO	G1 Zeichensatz einstellen
SI	G0 Zeichensatz einstellen

A.1.5 Darstellungsattribute

ESC [0m	normal (kein Attribut aktiv)
ESC [1m	fett (dargestellt in hervorgehobener Farbe)
ESC [4m	unterstrichen
ESC [5m	blinkend
ESC [7m	invertiert

A.1.6 Scrolling Region

ESC [Pt;Pbr	Scrolling Region setzen
-------------	-------------------------

A.1.7 Cursorsteuerung

ESC [PnA	Cursor Pn Zeilen nach oben
ESC [PnB	Cursor Pn Zeilen nach unten
ESC [PnC	Cursor Pn Spalten nach rechts
ESC [PnD	Cursor Pn Spalten nach links
ESC [Pl;PcH	Cursorposition setzen
ESC [Pl;Pcf	
ESC D	Index (8bit IND) (1 Zeile nach unten)
ESC M	Umgekehrter Index (8bit RI) (1 Zeile nach oben)
ESC E	Nächste Zeile (8bit NEL)
ESC 7	Cursorposition speichern
ESC 8	Cursorposition zurückladen

A.1.8 Tabulatur

ESC H	Horizontaltabulator setzen (8bit HTS)
-------	--

ESC [0g	Tabulator löschen
ESC [3g	Alle Tabulatoren löschen

A.1.1.9 Löschen

ESC [OK	Zeile von Cursorspalte bis zum Zeilenende
ESC [1K	Zeile vom Zeilenanfang bis zur Cursorspalte
ESC [2K	Gesamte Zeile

ESC [0J	Bildschirm von der Cursorzeile bis zur untersten Zeile
ESC [1J	Bildschirm von der obersten Zeile bis zur Cursorzeile
ESC [2J	Gesamter Bildschirm

A.1.1.10 Reports

	Abfrage vom Host		Antwort
ESC [5n	Status	ESC [0n	Terminal OK
		ESC [3n	Terminal nicht OK
ESC [6n	Cursorposition	ESC [Pl;Pcn	Cursorposition
ESC [c	Gerätetyp	ESC [?1;2c	VT100 mit Graphikoption
ESC [0c			
ESC Z			
ESC [?15n	Druckerstatus	ESC [?13n	Kein Drucker

A.1.1.11 Rücksetzen

ESC c	Reset auf Anfangswerte
-------	------------------------

A.1.1.12 Tastaturmodi

ESC =	Keypad Applikationsmodus
ESC >	Keypad numerischer Modus

A.1.1.13 Editierfunktionen

ESC [PnX	Pn Zeichen löschen
ESC [PnL	Pn Zeilen einfügen
ESC [PnM	Pn Zeilen löschen

A.1.14 Steuerzeichen (7 bit und 8 bit)

ENQ	05 Answerback (wird ignoriert)
BEL	07 Bell (Tonsignal)
BS	08 Backspace
HT	09 Horizontaltabulator
LF	0A Zeilenvorschub oder CR LF (bei Autowrap)
VT	0B wie LF
FF	0C wie LF
CR	0D Zeilenrücklauf
SO	0E Umschalten in den G1 Zeichensatz
SI	0F Umschalten in den G0 Zeichensatz
DLE	10 GKS Modus
DC1	11 XON
DC3	13 XOFF
CAN	18 Cancel (Textterminalmodus)
SUB	1A Cancel
ESC	1B Kontrollsequenzeinleitung
GS	1D 4014 Vektormodus
DEL	7F Zeichen löschen
IND	84 Index (ESC D)
NEL	85 Next line (ESC E)
HTS	88 Horizontal Tab setzen (ESC H)
RI	8D Reverse index (ESC M)
DCS	90 Device control string Einleitung (ESC P)
CSI	9B Kontrollsequenz Einleitung (ESC [)
ST	9C String terminator (ESC \)

Alle nicht genannten Steuerzeichen werden ignoriert.

A.2 Tektronix 4014/Retrographics 640 Modus

A.2.1 Alpha-Modus

Cursorbewegung

BS	Cursor nach links
HT	Cursor nach rechts
LF	Cursor nach unten
VT	Cursor nach oben
CR	Cursor an den aktuellen linken Rand

Zeichensatzgröße

ESC 0	Normalgröße 35x76 (Zeilen x Spalten)
ESC 1	Größe * 2
ESC 2	Größe * 3
ESC 3	Größe * 4

Moduswechsel

GS	Vektormodus
ESC SUB	GIN modus (Eingabe)
FS	Punkt plot modus
RS	Inkrementalmodus
US	Alphamodus, Keine Bewegung
CR	Alphamodus, Carriage Return
ESC FF	Alphamodus, Bildschirm löschen
CAN	Textterminalmodus

A.2.2 Sonstiges

ESC ENQ	Statusabfrage
ESC /0d	Dots an
ESC /1d	Dots aus
ESC /2d	Dots komplementär

A.2.3 Vektor Mode

GS	Nächster Vektor wird nicht gezeichnet
adresse	Zeichne Vektor, nächster Vektor sichtbar
ESC ‘	Durchgezogen
ESC a	Gepunktet
ESC b	Strichpunktiert
ESC c	Kurz-gestrichelt
ESC d	Lang-gestrichelt

A.2.4 Punkt Plot Modus

adresse	Punkt setzen
---------	--------------

Die Adresse muß in 12 bit Tektronix Kodierung angegeben werden.

A.2.5 Inkremental Modus

SP	Pen up
P	Pen down
D	Nord
E	Nordost
A	Ost
I	Südost
H	Süd
J	Süd west
B	West
F	Nordwest

A.3 Nonstandard Funktionen

A.3.1 Grundeinstellung

ESC [0;0Y	Linefeed-Modus: CR = CR , LF = LF
ESC [0;1Y	Linefeed-Modus: CR = CR , LF = CR LF
ESC [0;2Y	Linefeed-Modus: CR = CR LF , LF = LF
ESC [1;sx;sy;x0;y0Y	Tektronix-Schirm Lage und Größe: sx, sy Größe Pixel, x0, y0 Startpunkt
ESC [77;PnY	GKS-Modus auf Pn: 0 = Aus, 1 = An (Eingaben auf Ethernet)
ESC [98;PnY	Lokalmodus auf Pn: 0=Aus, 1=Echo auf Konsole (Tastatur)
ESC [99;Pzx;PzyY	Zoomfaktor für Textschirm setzen

A.3.2 Graphikkommandos

ESC [2;X;Y;OpmY	Absolut Dot an Punkt (X,Y), Opm: 0=Reset, 1=Set, 2=Komplement
ESC [3;X1;Y1;X2;Y2;Typ;OpmY	Absolut Linie von (X1,Y2) nach (X2,Y2) mit Linetype Typ
ESC [4;X;YY	Absolut Move auf (X,Y)
ESC [5;dX;dYY	Relativ Move um (dX,dY)
ESC [6;X;Y;OpmY	Absolut Linie nach Punkt (X,Y)
ESC [7;dX;dY;OpmY	Relativ Linie mit Größe (dX,dY)
ESC [8;X;Y;OpmY	Absolut Rechteck nach Punkt (X,Y)
ESC [9;dX;dY;OpmY	Relativ Rechteck mit Größe (dX,dY)
ESC [10;X;Y;OpmY	Absolut gefülltes Rechteck nach Punkt (X,Y)
ESC [11;dX;dY;OpmY	Relativ gefülltes Rechteck mit Größe (dX,dY)
ESC [12;R;OpmY	Kreis mit Radius (R)

ESC [13; A; B; dX; OpmY	Ellipse mit Parametern (A,B,dx)
ESC [14; Xc; Yc; Xe; Ye; OpmY	Absolut Kreisbogen von Punkt (Xc,Yc) nach Punkt (Xc,Yc)
ESC [15; Xc; Yc; Xe; Ye; OpmY	Relativ Kreisbogen von Punkt (Xc,Yc) nach Punkt (Xc,Yc)
ESC [16; OpmY	Paint

A.3.3 Graphikparameter

ESC [47; Fg; BgY	Farbindex Vordergrund Fg, Hintergrund Bg
ESC [48; Pt; SzY	Schreibpattern: Bitmuster Pt, Größe (bits) Sz
ESC [49; CtlY	ACRTC-Auswahl: 0 = ACRTC 0, 1 = ACRTC 1, 2 = beide
ESC [51; Fg; Bg; Edg; Pt; Sz; CtlY	Farbindex, Schreibpattern, ACRTC-Auswahl in einem

A.3.4 Beeinflussung Farbpalette

ESC [50; I; R; G; BY	Farbpaletteneintrag Index I, Farbton setzen auf R, G, B, (je 0-255)
ESC [60; IY	Farbpaletteneintrag Index I holen, Antwort: 60; R; G; B CR

A.3.5 Beeinflussung Mouse Pointer

ESC [52; MpatY	Pointer (mouse) Pattern setzen (0-5)
ESC [53; X; Y	Pointerposition auf (X,Y) setzen
ESC [63; IY	Pointerposition holen, Antwort: X; Y CR
ESC [54Y	Prompt-Echo-Modus Gummibandrechteck setzen

B Dokumentation Graphiksystemsoftware

B.1 Kontrollsequenzen der GKS-Anbindung

DLE <i>Cabcde</i> CR	Create Window Id a (0-15), unten links (b,c) Größe (d,e)
DLE <i>Da</i> CR	Delete Window Id a (0-15)
DLE <i>Ea</i> CR	Erase Window Id a (0-15)
DLE <i>Labcdefxyy</i> CR	Draw Lines in Window Id a (0-15)
DLE <i>Pabcdefxyy</i> CR	Polyline in Window Id a (0-15)
DLE <i>Tabcdefghij</i> LF <i>text</i> CR	Text in Window Id a (0-15)
DLE <i>Fabcdefxyy</i> CR	Polygon Fill in Window Id a (0-15)
DLE <i>Aabcdefgh</i> CR	Circle in Window Id a (0-15)
DLE <i>Rirgb</i> CR	Set Color Representation
DLE <i>Bab</i> CR	Set Mouse Pattern
DLE <i>Habcd</i> CR	Set Prompt Echo Type
DLE <i>Maxy</i> CR	Set Mouse Position relativ zu Window a
DLE <i>Ga</i> CR	Get Mouse Position relativ zu Window a
DLE <i>Xabcdef</i> CR	Backing Store a

B.2 Parameterkodierung

Parameterkodierung: Die Parameter werden je in 2 Zeichen mit je 6 bit und Offset 32 aufgeteilt. Die dadurch übertragenen Werte liegen im Bereich von 32...95.

B.3 Erklärung der GKS-Kommandos

Create Window: (Funktionsname G\$CREATE) Baut ein Fenster in der angegebenen Größe auf. Parameter:

- *Wd_Id*: Window Id (Wertebereich 0...15). Identifikationsnummer des aufgebauten Fenster, unter der folgende Kommandos auf dieses Fenster wirken.
- *X_Pos*: X-Position (0...1279). X-Koordinate der unteren linken Ecke des Fensters. (0,0) ist auf dem Schirm unten links, (1279,1023) oben rechts.
- *Y_Pos*: Y-Position (0...1023). Y-koordinate der unteren linken Ecke.
- *X_Siz*: X-Size (0...1279). Größe des aktiven Bereichs des Fensters in X-Richtung.
- *Y_Siz*: Y-Size (0...1023). Größe des aktiven Bereichs des Fensters in Y-Richtung.

Delete Window: (G\$DELETE) Löscht das Fenster mit der angegebenen Identifikation vom Bildschirm. Das Fenster ist danach nicht mehr vorhanden. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters.

Erase Window: (G\$ERASE) Löscht den Inhalt des Fenster mit der angegebenen Identifikation. Das Fenster ist danach weiterhin ansprechbar. vorhanden. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters.

Draw Line: (G\$LINE) Zeichnet gerade Linien mit den angegebenen Attributen zwischen jeweils zwei der angegebenen Koordinatenpaare. Beispiel: Bei L=4 von (x1,y1) nach (x2,y2) und von (x3,y3) nach (x4,y4). Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, in das gezeichnet wird.
- WrMod: Write Mode (1...3). Schreibmodus: 1=Löschen, 2=Überschreiben 3=Invertieren.
- WrInd: Write Index (0...15). Farbindex der Farbe, mit der geschrieben wird.
- L_Sty: Line Style (0...3). Linientyp: 0=Durchgezogen, 1=Gestrichelt, 2=Gepunktet, 3=Strichpunktet.
- L_Wid: Line Width (1...4095) Linienbreite in Bildpunkten. Wird durch das Zeichen von mehreren versetzten Linien simuliert.
- N: Parameter Count (1...497) Anzahl der folgenden Koordinaten.
- X: X Value (0...1279) X-Koordinatenwert. Dieses Feld muß N mal vorhanden sein. Die Koordinaten sind relativ zum Ursprung des Fenster (untere linke Ecke des Fensters) und werden bei Überschreiten der Fenstergröße geclippt.
- Y: Y Value (0...1023) Y-Koordinatenwert. Auch dieses Feld wird N mal wiederholt.

Polyline: (G\$PLOT) Zeichnet einen offenen Linienzug mit den angegebenen Attributen zwischen den angegebenen Koordinatenpaaren. Beispiel: Bei L=4 von (x1,y1) nach (x2,y2), von hier nach (x3,y3), von hier nach (x4,y4). Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, in das gezeichnet wird.
- WrMod: Write Mode (1...3). Schreibmodus: 1=Löschen, 2=Überschreiben 3=Invertieren.
- WrInd: Write Index (0...15). Farbindex der Schreibfarbe.

- L_Sty: Line Style (0...3). Linientyp: 0=Durchgezogen, 1=Gestrichelt, 2=Gepunktet, 3=Strichpunktet.
- L_Wid: Line Width (1...4095) Linienbreite in Bildpunkten. Wird durch das Zeichen von mehreren versetzten Linien simuliert.
- N: Parameter Count (1...497) Anzahl der folgenden Koordinaten.
- X: X Value (0...1279) X-Koordinatenwert. Dieses Feld muß N mal vorhanden sein. Die Koordinaten sind relativ zum Ursprung des Fenster (untere linke Ecke des Fensters) und werden bei Überschreiten der Fenstergröße geclippt.
- Y: Y Value (0...1023) Y-Koordinatenwert. Auch dieses Feld wird N mal wiederholt.

Text: (G\$TEXT) Ausgabe der Zeichenkette in der gewählten Richtung, Größe und Farbe. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, in das gezeichnet wird.
- WrMod: Write Mode (1...3). Schreibmodus: 1=Löschen, 2=Überschreiben 3=Invertieren.
- WrInd: Write Index (0...15). Farbindex der Schreibfarbe.
- Font: Font Index (1...11). Nummer des gewählten Fonts. Die Fonts unterscheiden sich durch ihre Höhe, Breite und den Abstand der Zeichen voneinander.
- C_Hei: Character Height (0...255). Höhe des Zeichen in Pixeln. Die Höhe wird durch Zoomen des gewählten Fonts in 16 Stufen von je 16 Pixeln angenähert.
- C_Wid: Character Width (0...127). Breite des Zeichen in Pixeln. Die Breite wird durch Zoomen des gewählten Fonts in 16 Stufen von je 8 Pixeln angenähert.
- C_Spc: Character Spacing (0...4095). Zusätzlicher Abstand zwischen den einzelnen Zeichen einer Zeichenkette.
- C_Slo: Character Slope (0...359). Steigung der Basislinie der Zeichenkette in Grad. 0 Grad entspricht dem Schreiben nach genau rechts. Die Zeichenneigung wird in 8 Stufen angenähert, die Positionierung ist exakt.
- X: X Start (0...1279) X-Startpunkt des ersten Zeichens.
- Y: Y Start (0...1023) Y-Startpunkt des ersten Zeichens.
- Text: Text. Die Zeichenkette wird durch ein Line Feed (hex 0A) eingeleitet und mit Carriage Return (hex 0D) abgeschlossen. Die Zeichen werden byteweise und nicht kodiert übergeben. Erlaubt sind alle darstellbaren ASCII- Werte (32...127).

Polygon Fill: (G\$FILL) Füllt den Innenraum eines geschlossenen Linienzug mit den angegebenen Attributen zwischen den angegebenen Koordinatenpaaren. Das Polygon darf sowohl konvex als auch konkav sein und die eigenen Linien schneiden. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, in das gezeichnet wird.
- WrMod: Write Mode (1...3). Schreibmodus: 1=Löschen, 2=Überschreiben 3=Invertieren.
- WrInd: Write Index (0...15). Farbindex der Schreibfarbe.
- F_Mod: Fill Mode (0...3). Füllmodus: 0=Hohl (zieht nur das Polygon), 1=Gefüllt, 2=User Pattern (wie 0), 3=Schraffiert.
- F_Pat: Fill Pattern (1...58) Füllpatternindex bei Füllmodus 3. Es stehen 58 verschiedene Schraffurtypen zur Auswahl, die denen von UIS entsprechen.
- N: Parameter Count (1...497) Anzahl der folgenden Koordinaten.
- X: X Value (0...1279) X-Koordinatenwert. Dieses Feld muß N mal vorhanden sein.
- Y: Y Value (0...1023) Y-Koordinatenwert. Auch dieses Feld wird N mal wiederholt.

Circle: (G\$CIRCLE) Zeichnet einen Kreis mit Radius R um das angegebene Koordinatenpaar mit den angegebenen Attributen. Bei Überschreiten der Fenstergrenzen wird der Kreis geklippt. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, in das gezeichnet wird.
- WrMod: Write Mode (1...3). Schreibmodus: 1=Löschen, 2=Überschreiben 3=Invertieren.
- WrInd: Write Index (0...15). Farbindex der Schreibfarbe.
- L_Sty: Line Style (0...3). Linientyp: 0=Durchgezogen, 1=Gestrichelt, 2=Gepunktet, 3=Strichpunktet.
- L_Wid: Line Width (1...4095) Linienbreite in Bildpunkten. Wird durch das Zeichen von mehreren unterschiedlich großen Kreisen simuliert.
- X: X Value (0...1279). X-Wert des Mittelpunkt des Kreises.
- Y: Y Value (0...1023). Y-Wert des Mittelpunkts des Kreises.
- R: Radius (0...4095). Radius des Kreises.

Set Color Representation: (G\$COLOR) Setzt den Farbton für einen Farbindex in der Farbpalette auf den angegebenen Wert. Parameter:

- Index: Index (0...15). Farbindex
- Red: Red Intensity (0...15). Intensität des Rotanteils.
- Grn: Green Intensity (0...15). Intensität des Grünanteils.
- Blu: Blue Intensity (0...15). Intensität des Blauanteils.

Set Mouse Pattern: (G\$MOUSEPAT) Setzt die Darstellung des Mauszeiger auf den angegebenen Wert. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, auf das sich der Befehl bezieht (keine fensterspezifische Darstellung implementiert).
- Pat: Pattern Index (0...5). Index auf die Mauszeigerform.

Set Mouse Position: (G\$MOUSEPOS) Setzt die Position des Mauszeiger auf das angegebene Koordinatenpaar. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, auf das sich der Befehl bezieht (wird zur Koordinatentransformation verwendet).
- X: X Value (0...1279). X-Wert des Mauszeigers.
- Y: Y Value (0...1023). Y-Wert des Mauszeigers.

Get Mouse Position: (G\$GETMPOS) Sendet die Position des Mauszeiger in fensterrelativen Koordinaten an den Host. Die Koordinaten werden wie die Leseparameter codiert. Erst wird in 2 Bytes (LSB/MSB) die X-Koordinate, dann in 2 Bytes die Y-Koordinate gesendet. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, auf das sich der Befehl bezieht (wird zur Koordinatentransformation verwendet).

Set Prompt Echo Type: (G\$PROMPTECHO) Setzt die Initialposition und die Art des Echo für die Zeigereingabe. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, auf das sich der Befehl bezieht (wird zur Koordinatentransformation verwendet).
- PET: Prompt Echo Type (0...5). Index auf den Echo-Typ. 2=Crosshair, 4=Gummiband, 5=Gummibandrechteck, 0,1,3=Mauszeiger
- X: X Value (0...1279). X-Initialwert.
- Y: Y Value (0...1023). Y-Initialwert.

Backing Store: (G\$BACK) Zwischenspeichern eines rechteckigen Ausschnitts im unsichtbaren Teil des Videospeichers zum späteren Restaurieren. Dieser Befehl wird für das Abspeichern des Hintergrund bei einer Auswahl-Eingabe mit Menubox verwendet. Die maximale Höhe ($Y_{max}-Y_{min}$) beträgt 560 Pixel. Größere Werte werden ignoriert. Parameter:

- Wd_Id: Window Id (0...15). Identifikationsnummer des Fensters, auf das sich der Befehl bezieht (wird zur Koordinatentransformation verwendet).
- Mode: Modus (1...2). 1=Abspeichern des Inhalts und Ausfüllen des Bereichs mit Farbindex 9, 2=Restaurieren des Bereichs.
- Xmin: X Minimum (0...1279). X-Koordinate des unteren rechten Punktes des Ausschnitts.
- Xmax: X Maximum (0...1279). X-Koordinate des oberen linken Punktes des Ausschnitts.
- Ymin: Y Minimum (0...1023). Y-Koordinate des unteren rechten Punktes des Ausschnitts.
- Ymax: Y Maximum (0...1023). Y-Koordinate des oberen linken Punktes des Ausschnitts.

C Dokumentation GKS-Treiber

C.1 Modulstruktur

GKY46A - Hauptmodul

GK46xx - Untermodule:

GK46AA - Acknowledgement

GK46AC - Configuration

GK46AD - Delete Prompt Echo

GK46AF - Get Window Size (aus File WINDOW.DAT)

GK46AG - Generate Prompt Echo

GK46AH - Queue Handler (Input Queue)

GK46AJ - Input Handler AST (Ethernet Transport)

GK46AM - Manipulate Prompt Echo

GK46AO - Output Handler

GK46AP - Get Pointer Position

GK46AR - Report Device Facilities

GK46AS - Generate Control Sequences

GK46AW - Buffer Handler (Output Queue)

GK46ET - Ethernet Transport Handler (C)

C.2 Gerätebeschreibungstabelle

Die Gerätebeschreibung entspricht im wesentlichen der Gerätebeschreibung für den DEGPX-Treiber. Die Unterschiede sind:

- **Buffer Mode:** Der GKS-Treiber buffert alle Ausgaben zwischen und erlaubt so einen höheren Protokolldurchsatz.
- **Workstations:** Die Workstationstypen sind von 8700 bis 8715 durchnummeriert. Maximal werden 16 Workstations dargestellt.
- **Eingabe:** Die Eingabe wird immer der Workstation zugeordnet, in deren Window sich der Mouse Pointer befindet.
- **Füllmodus:** Beim Polygonfill werden geringfügig unterschiedliche Punktmen- gen in y-Richtung gesetzt, da die Rundung des Polygons nach verschiedenen Verfahren vorgenommen wird.
- **Window Priorität:** Neue Windows werden immer oben auf die schon vorhande- nen Windows gesetzt. Dazu wird der von anderen Windows belegte Hintergrund zwischengespeichert. Dazu müssen alle unterliegenden Windows in der umge- kehrten Reihenfolge ihrer Priorität an die Oberfläche geholt werden. Das kann

bei vielen übereinanderliegenden Windows eine gewisse Zeit dauern. Ist wird daher empfohlen, so wenig wie möglich übereinanderliegende Windows zu erzeugen. Teilweise verdeckte Windows können durch Anklicken mit der rechten Maustaste nach oben geholt werden. Völlig verdeckte Windows können durch wiederholtes Drücken der Funktionstaste F2 wieder nach oben geholt werden. Wenn der Hintergrundspeicher nicht mehr ausreicht, können Windows verloren gehen. Die Größe des Hintergrundspeichers beträgt 1 Megabyte. Eine Erweiterung auf 4 Megabyte ist bei Bedarf möglich.

Literaturverzeichnis

- [1] H.-D. Gräf, H. Miska, E. Spamer, O. Titze and Th. Walcher, Nucl. Instr. Meth. **153** (1978) 9;
Th. Walcher, R. Frey, H.-D. Gräf, E. Spamer and H. Theissen, Nucl. Instr. Meth. **153** (1978) 17;
D. Schüll, J. Foh, H.-D. Gräf, H. Miska, R. Schneider, E. Spamer, H. Theissen, O. Titze and Th. Walcher, Nucl. Instr. Meth. **153** (1978) 29;
J. Foh, R. Frey, R. Schneider, A. Schwierczinski, H. Theissen and O. Titze, Nucl. Instr. Meth. **153** (1978) 43
- [2] K. Alrutz-Ziemssen, D. Flasche, H.-D. Gräf, V. Huck, M. Knirsch, W. Lotz, A. Richter, T. Rietdorf, P. Schardt, E. Spamer, A. Staschek, O. Titze, W. Voigt, H. Weise and W. Ziegler, Proc. 4th Workshop on RF-Superconductivity, Tsukuba, Japan, KEK **89-21** (1989) 53;
K. Alrutz-Ziemssen, D. Flasche, H.-D. Gräf, V. Huck, M. Knirsch, W. Lotz, A. Richter, T. Rietdorf, P. Schardt, E. Spamer, A. Staschek, W. Voigt, H. Weise and W. Ziegler, Proc. Part. Acc. Conf. **29** (1990) 53
- [3] A. Richter, Phys. Bl. **43** (1987) 452
- [4] Cern Courier **27** (1987) 7, 34
- [5] V. Aab, K. Alrutz-Ziemssen, H. Genz, H.-D. Gräf, A. Richter, H. Weise and A. Gaupp, Proc. 10th Int. Free Electron Laser Conference, Jerusalem, Israel, Nucl. Instr. Meth. **A285** (1989) 76
- [6] M. Knirsch, Dissertation, TH Darmstadt, in Vorbereitung
- [7] K.D. Hummel, Dissertation, TH Darmstadt, in Vorbereitung
- [8] A. Stiller, Diplomarbeit, TH Darmstadt (Februar 1991), unveröffentlicht
- [9] G. Herbert, Dissertation, TH Darmstadt, in Vorbereitung
- [10] G. Kalisch, Diplomarbeit, TH Darmstadt (Oktober 1990), unveröffentlicht
- [11] K. Alrutz-Ziemssen, Diplomarbeit, TH Darmstadt (Mai 1986), unveröffentlicht
- [12] W. Ziegler, Dissertation, TH Darmstadt (Februar 1991)
- [13] W. Lotz, Dissertation, TH Darmstadt (Juli 1990)
- [14] M. Schanz, Dissertation, TH Darmstadt (Oktober 1986)
- [15] M. Knirsch, Diplomarbeit, TH Darmstadt (November 1986), unveröffentlicht
- [16] F. Lindqvist, Diplomarbeit, TH Darmstadt (April 1988), unveröffentlicht

- [17] C. Lüttge, private Mitteilung (1991)
- [18] H. Diesener, private Mitteilung (1991)
- [19] Particle Data Group, **Review of Particle Properties**,
Physics Letters **B239** (April 1990)
- [20] G. Kuchler, Dissertation, TH Darmstadt (Oktober 1986)
- [21] F. Sauli, Principles of Operation of Multiwire Proportional and Drift Chambers,
CERN **77 - 09** (Mai 1977)
- [22] K. D. Hummel, private Mitteilung (1991)
- [23] A. Steinmetz, Diplomarbeit, TH Darmstadt (Juli 1990), unveröffentlicht
- [24] W. Bertozzi, M.V. Hynes, C.P. Sargent, C. Creswell, P.C. Dunn A. Hirsch, M.
Leitch, B. Norum, F.N. Rad and T. Sasanuma, Nucl. Instr. Meth. **141** (1977) 457
- [25] W. Bertozzi, M.V. Hynes, C.P. Sargent, W. Turchinetz and C. Williamson,
Nucl. Instr. Meth. **162** (1979) 211
- [26] K.Kleinknecht, **Detektoren für Teilchenstrahlung**,
Teubner, Stuttgart (1987)
- [27] G.A. Erskine, Nucl. Instr. Meth. **105** (1972) 565
- [28] T. Gallenkamp, Studienarbeit, TH Darmstadt (Juni 1988), unveröffentlicht
- [29] GKS - International Standard ISO IS 7942 (1985)
- [30] R.W. Scheifler, J. Gettys, R. Newman,
X Window System, Digital Press (1988)
- [31] G. Enderle, K. Kansy, G. Pfaff,
Computer Graphics Programming, Springer,
Berlin Heidelberg New York Tokyo (1984)
- [32] GKSGRAL User And Reference Manual,
GTS-GRAL GmbH, Darmstadt (1987)
- [33] GKSGRAL Device Driver Reference Manual
GTS-GRAL GmbH, Darmstadt (1989)
- [34] VT330/VT340 Programmer Pocket Guide, DEC (1987)
- [35] HD 63484 ACRTC Advanced CRT Controller User Manual, Hitachi (1984)

Danksagung

Mein Dank gilt an dieser Stelle all denen, die am Zustandekommen dieser Arbeit beteiligt waren und meinen Aufenthalt am Institut für Kernphysik lehr- und abwechslungsreich gestalteten.

Zu allererst danke ich Herrn Professor Dr. A. Richter, der durch seine interessante Themenstellung mein Interesse geweckt hat. Durch seine motivierende Diskussionsbereitschaft hat er wesentlichen Anteil am Gelingen dieser Arbeit.

Herrn Dr. G. Schrieder und Herrn Dr. O. Titze danke ich für die Betreuung und für die Unterstützung beim Anfertigen dieser Arbeit.

Für ihre Diskussionsbereitschaft und die engagierte Unterstützung bei den vielfältig auftretenden Fragen danke ich den Herren H. Diesener, Dr. D. Flasche, K.D. Hummel, G. Herbert, M. Knirsch, C. Lüttge, A. Steinmetz, A. Stiller, J. Töpfer, H. Weise und H.J. Wörtche.

Hiermit erkläre ich, daß ich die vorliegende Diplomarbeit selbständig verfaßt und nur die angegebenen Hilfsmittel verwendet habe.

Darmstadt, den 3. Mai 1991