

Konzeption eines
wissenschaftspropädeutischen Seminars zum
Thema kosmische Strahlung

ZULASSUNGSARBEIT

Im Studiengang

Lehramt an Gymnasien für die Fächerkombination
Physik und Mathematik

der

Friedrich-Alexander-Universität
Erlangen-Nürnberg
in der Naturwissenschaftlichen Fakultät und Fachbereich Physik

Betreuer:

Herr Prof. Dr. Christian Stegmann

vorgelegt von:

Friedrich Meingast aus Nördlingen

Erlangen, im September 2010

Inhaltsverzeichnis

Tabellenverzeichnis	vi
Abbildungsverzeichnis	vii
Abkürzungsverzeichnis	ix
1 Einleitung und Ziele	1
2 Der Cosmic Ray Muon Detector	4
2.1 Aufbau	4
2.2 Funktion	9
2.2.1 Input	9
2.2.2 Timing	11
2.2.3 Koinzidenzbedingungen	12
2.2.4 Output	13
2.3 Installation von Treibern und Software	15
2.3.1 Treiber für USB-Schnittstelle	15
2.3.2 Terminal-Software	15
2.3.3 Einstellung des Terminals	16
2.4 Kalibration	17
2.4.1 Bestimmung der Schwellenspannung	17
2.4.2 Bestimmung der Spannungsversorgung im Single-Modus	18
2.4.3 Bestimmung der Spannungsversorgung im Koinzidenz-Modus	19
2.4.4 Optimierung	20
3 Versuche mit dem Cosmic Ray Muon Detector	21
3.1 Analyse der Leistungsfähigkeit des Detektorsets	21
3.1.1 Aufbau und Methoden	22
3.1.2 Daten und Analyse	22
3.1.3 Fehler	24
3.2 Messung des Myonenflusses	25
3.2.1 Aufbau und Methoden	25

Inhaltsverzeichnis

3.2.2	Daten und Analyse	26
3.2.3	Fehler	29
3.3	Messung der Myonen-Lebensdauer	31
3.3.1	Aufbau und Methoden	32
3.3.2	Daten und Analyse	34
3.3.3	Fehler	38
3.4	Schauerexperimente	40
3.4.1	Aufbau und Methoden	40
3.4.2	Daten und Analyse	42
3.4.3	Fehler	48
3.5	Weitere Experimente	51
3.5.1	Abhängigkeit der Schauerrate von der Detektoranordnung . . .	51
3.5.2	Geschwindigkeitsmessung	52
3.5.3	Vermessung der Größe des Mondschattens	52
4	Java-Werkzeuge zur Datennahme und Auswertung	54
4.1	Kalibration mit autocal	55
4.2	Datennahme und Bearbeitung	56
4.2.1	getdata	56
4.2.2	linecheck	56
4.2.3	fill	57
4.2.4	correct	58
4.2.5	sort	59
4.3	Auswertung	60
4.3.1	readout	60
4.3.2	gnutot	61
4.3.3	rate	61
4.3.4	anglet	62
4.3.5	histo	62
4.4	Datenformate	64
4.4.1	Rohdaten-Format	64
4.4.2	readout-Format	64
4.4.3	gnutot-Format	65
4.4.4	rate-Format	65
4.4.5	config-Format	65
4.4.6	anglet-Format	66
4.4.7	histo-Format	66
5	Array-Studie	67
5.1	Positionen der Schulen in Erlangen	67

Inhaltsverzeichnis

5.2	Simulationsuntersuchungen zur Messung der kosmischen Strahlung mit Szintillationsdetektoren	69
6	Wissenschaftspropädeutisches Seminar zur kosmischen Strahlung	72
6.1	Vorgaben	72
6.1.1	Ziel: Kompetenz in wissenschaftlichem Arbeiten	72
6.1.2	Leitfach, Lehrereinsatz und Rahmenthema	73
6.1.3	Methodik und Zeitplan	74
6.1.4	Lernort Bibliothek	75
6.1.5	Leistungserhebung	76
6.1.6	Konzepterstellung durch die Lehrkraft	77
6.1.7	Wahlprozess zur Einrichtung der Seminarkurse	77
6.1.8	Maßnahmen zur Qualitätssicherung	79
6.2	Umsetzung	80
6.2.1	Ziel: Kompetenz in wissenschaftlichem Arbeiten	80
6.2.2	Leitfach, Lehrereinsatz und Rahmenthema	80
6.2.3	Methodik und Zeitplan	80
6.2.4	Lernort Bibliothek	81
6.2.5	Leistungserhebung	82
6.2.6	Konzepterstellung durch die Lehrkraft	83
6.2.7	Wahlprozess zur Einrichtung der Seminarkurse	85
6.2.8	Maßnahmen zur Qualitätssicherung	85
6.3	Vorschlag zum Aufbau eines Lehrplans	86
6.4	Literatur für Schüler zum Thema kosmische Strahlung	88
6.4.1	Printmedien	88
6.4.2	Online-Medien	88
6.4.3	Literaturliste	89
7	Organisation	92
7.1	Lehrerfortbildung - Astroteilchenphysik	92
7.2	Betreuung Detektorbau	93
7.3	Betreuung Detektorarray	93
8	Fazit	94
A	Board-Befehle	96
B	Java-Quelltexte	99
B.1	autocal	99
B.2	getdata	101
B.3	linecheck	103

Inhaltsverzeichnis

B.4 fill	105
B.5 correct	106
B.6 sort	108
B.7 readout	109
B.8 gnutot	111
B.9 rate	112
B.10 angle	113
B.11 anglet	118
B.12 histo	120
Literaturverzeichnis	123
Wahrheitsgemäße Erklärung	126

Tabellenverzeichnis

2.1	Wichtige Steuerbefehle	10
2.2	Bedeutung der Rohdaten	14
2.3	Parameter zur Einstellung des Terminals	16
3.1	Datensätze zur Messung der Myonenlebensdauer	34
3.2	Ergebnisse für die Myonenlebensdauer mit Fitmethode 1	35
3.3	Ergebnisse für die Myonenlebensdauer mit Fitmethode 2	37
5.1	Positionen der Erlanger Gymnasien ermittelt mit GoogleMaps	68
5.2	Zählraten $[1/m^2 \cdot a]$ an Stelle des ECAP und der drei nächstgelegenen Schulen	70
5.3	Wahrscheinlichkeiten für mindestens ein Ereignis in 90 Tagen	71

Abbildungsverzeichnis

1.1	Zahl der Studienanfänger im Fach Physik an der FAU-Erlangen-Nürnberg [Fri09]	3
2.1	Verpacken der Szintillatorplatte	5
2.2	Lichtdicht verpackte Szintillatorplatte	6
2.3	Anbringung des PMT an den Szintillator	7
2.4	Fertiggestelltes Detektor-„Paddel“	8
2.5	Schaltbild des DAQ-Boards [RJPB10]	9
2.6	Zusammenhang von TL, RE, FE, TOT (GnuPlot)	11
2.7	Messung der der Zählrate in Abhängigkeit der Schwellenspannung für einen Kanal (GnuPlot)	17
2.8	Messung der Zählrate in Abhängigkeit der Betriebsspannung im Single-Modus für einen Kanal (GnuPlot)	18
2.9	Anordnung zur Kalibration im Koinzidenz-Modus	19
2.10	Messung der Zählrate in Abhängigkeit der Betriebsspannung im Koinzidenz-Modus für einen Kanal (GnuPlot)	20
3.1	Beziehung zwischen Signal und TOT (GnuPlot)	22
3.2	Startseite der Online-Plattform „Cosmic Ray e-Lab“	23
3.3	Analyse der TOT (GnuPlot)	23
3.4	Möglicher Aufbau zur Myonenfluss-Messung	25
3.5	Myonenrate am 21./22. April 2010 (CReLab)	26
3.6	Daten einer fehlerhaften Myonenfluss-Messung (12. Mai 2010, Set II, Kanal 1)	27
3.7	Gegenseitige Beeinflussung der Detektoren bei der Datenaufzeichnung (Daten vom 24. April 2010, Set II) (CReLab)	28
3.8	Unnatürliche Ratenveränderung in den Daten vom 7. März 2010 (Set II, Kanal 1) (CReLab)	29
3.9	Anordnung der Detektoren zur Messung der Myonenlebensdauer (GnuPlot)	32
3.10	Gestapelte Detektoren zur Messung der Myonenlebensdauer	32

Abbildungsverzeichnis

3.11	Standorte der Messaufbauten (Google Maps)	33
3.12	Einstellung der Parameter zur Myonen-Lebensdauer-Messung mit dem „CR e-Lab“	35
3.13	Datensatz 3 mit 80 bins (GnuPlot)	36
3.14	Datensatz 1 mit 120 bins (GnuPlot)	38
3.15	Möglichkeiten der Detektoranordnung (GeoGebra)	40
3.16	Aufbau 1 (Google Maps)	41
3.17	Detektoranordnungen Aufbau 1 (GnuPlot)	42
3.18	Aufbau 2 (Google Maps)	43
3.19	Detektoranordnungen Aufbau 2 (GnuPlot)	44
3.20	Zusammenhang zwischen Schauerwinkel ϕ und Δt (GeoGebra)	44
3.21	Richtungsverteilung der Luftschauer gemessen mit Detektorset II (Gnu- Plot)	47
3.22	Winkelabhängigkeit der Schauerhäufigkeit (GnuPlot)	48
3.23	Rekonstruktion von Azimut und Höhe der 30001 gemessenen Schauer (GnuPlot)	49
3.24	Detektoranordnung in Abhängigkeit von R (GnuPlot)	51
3.25	Möglicher Aufbau zur Messung der mittleren Myonengeschwindigkeit (GnuPlot)	52
5.1	Schulen in Erlangen (GoogleMaps)	67
6.1	Einrichtung der W-Seminare - Phase 1 [MR08]	78
6.2	Einrichtung der W-Seminare - Phase 2 [MR08]	78
6.3	Formular zur Beantragung eines W-Seminars [MR08]	83

Abkürzungsverzeichnis

ADC	Analog-to-Digital-Converter Analog-Digital-Wandler
AMP	Amplifier Signalverstärker
COMP	Comparator (auch Discriminator) vergleicht Signal und gesetzte Schwellenwert
Cookie	runder Glasaufsatz an einer Ecke der Szintillatorplatte
CPLD	Complex-Programmable-Logic-Device programmierbares Logik-Modul
CRMD	Cosmic Ray Muon Detector Detektor für Myonen der kosmischen Strahlung
DAQ	Data Acquisition Board Karte zur Datenerfassung
Detektor	Einheit bestehend aus einem Szintillator, PMT und Verpackung
Detektorset	System aus vier Detektoren, DAQ-Board und GPS-Modul
ECAP	Erlangen Centre for Astroparticle Physics Erlanger Zentrum für Astroteilchen-Physik
FE	Falling Edge fallende Signalflanke
ISB	Institut für Schulqualität und Bildung in Bayern
MCU	Micro-Controller-Unit (Slow-Logic-Modul) für die Kommunikation via USB-Schnittstelle verantwortlicher Chip
PMT	Photo-Multiplier-Tube Sekundärelektronenvervielfacher in Röhrenform
QuarkNet	Netzwerk für Teilchenphysik-Projekte an Schulen in den USA
RE	Rising Edge ansteigende Signalflanke
TDC	Time-to-Digital-Converter Messkreis für kurze Zeitintervalle mit digitaler Ausgabe

Abbildungsverzeichnis

TL	Threshold-Level Schwellenspannung des Diskriminators, ab welcher ein Signal als solches interpretiert wird
TMC	Time-Measurement-Chip Zeitmessungschip
TOT	Time-over-Threshold Zeitspanne des Signals oberhalb des TL
W-Seminar	wissenschaftspropädeutisches Seminar

1 Einleitung und Ziele

„Physik ist das ungeliebteste Schulfach!“ [Mer08]

Zu diesem Ergebnis kam Gottfried Merzyn (Universität Göttingen) bei der Befragung von 750 SchülerInnen zur Beliebtheit und Unbeliebtheit der Unterrichtsfächer. Dabei gaben fast 50% der befragten Personen Physik als ungeliebtestes Fach an. Damit rangiert Physik noch vor Chemie und Mathematik. Im Rahmen der Erhebung wurde auch nach der Beliebtheit der Fächer gefragt. Die Physik landete hierbei im unteren Mittelfeld zwischen den Fächern Englisch und Musik. Merzyn schließt daraus, dass kein anderes Unterrichtsfach so stark polarisiert wie die Physik.

Trotz des alltäglichen Technik-Hypes um W-Lan, iPhone, E-Books, etc. ist Abneigung gegen Physik in der Gesellschaft verbreitet, obwohl diese vielgepriesenen Errungenschaften niemals ohne ihre Verdienste entwickelt worden wären. Merzyn begründet in [Mer09] die Ablehnung des Fachs Physik bei den SchülerInnen durch die Auswahl der Lehrinhalte, die weit von den Interessen der Kinder und Jugendlichen entfernt seien. Auch die am weitesten verbreitete Unterrichtsmethode, das fragend-entwickelnde Unterrichtsgespräch, kommt bei den SchülerInnen nicht gut an. Meist dominieren die LehrerInnen die Kommunikation und versuchen, die Lernenden auf den von ihnen vorgedachten „Gedankenpfad“ zu zwingen. Ein weiteres gravierendes Problem ist das Anforderungsniveau des Fachs Physik, das in der Schule zu den höchsten gehört. Das Scheitern daran frustriert die SchülerInnen und führt zu eben dieser hohen Ablehnungsquote des Fachs.

Mit dem Ziel wieder mehr SchülerInnen für die Physik zu begeistern, zeigt Merzyn neue Wege auf, um die eben erläuterten Probleme zu lösen. Die Inhaltsauswahl des Physikunterrichts soll sich stärker an den Alltagserfahrungen der SchülerInnen orientieren. Überdies soll sie sich auch nach Themen richten, die außerhalb der Schule wichtig sind und den Lernenden täglich in den Medien begegnen. Außerdem muss der Physikunterricht die Lehrerzentrierung aufgeben und sich einer weniger „engen“ Gesprächsführung zuwenden, die eigene Gedanken und Ideen zulässt. Die SchülerInnen

1 Einleitung und Ziele

sollten dazu auch mehr Chancen erhalten, selbst zu experimentieren und so ihren Erfahrungsschatz zu erweitern. Zudem muss das Anforderungsniveau so gewählt werden, dass die Kinder und Jugendlichen im Physikunterricht Erfolgserlebnisse haben. Nur so ist es möglich, ein nachhaltiges Interesse an der Physik und ihren Inhalten zu erzeugen. Im Vordergrund sollte dabei stets das Erlernen der charakteristischen Art des Denkens und Problemlösens in der Physik stehen.

Das in Bayern im Zuge der Oberstufenreform (G8) neu eingeführte „wissenschaftspropädeutische Seminar“ (**W-Seminar**) bietet mit seinen weit formulierten Rahmenbedingungen nun die Möglichkeit, losgelöst von allen Lehrplänen wissenschaftlich mit SchülerInnen zu arbeiten. Dabei können sämtliche Verbesserungspunkte von Merz syn berücksichtigt werden. Themen der Astronomie und Astroteilchenphysik lassen sich oft in den Medien wiederfinden. Diese Teilbereiche faszinieren die SchülerInnen, was auch durch das rege Interesse am „Jahr der Astronomie“ in jüngster Vergangenheit wiedergegeben wird. Besonders interessant für die Schule wird die Astroteilchenphysik, da sie sämtliche Größenordnungen von den Bausteinen der Materie bis zu den größten Strukturen des Universums umfasst [B⁺06]. Die Unterrichtsmethoden wechseln sich im W-Seminar ständig ab. Die SchülerInnen sollen selbstständiges Planen und Arbeiten erlernen und sich mit den spezifischen Methoden der Physik vertraut machen. Ziel ist es auch, schulübergreifende Projekte (z.B. zur Messung ausgedehnter Luftschauer) durchzuführen.

Abbildung 1.1 zeigt die Zahl der Studienanfänger im Fach Physik an der Friedrich-Alexander-Universität Erlangen-Nürnberg und leider auch den Abwärtstrend bei diesen. Die Chance der Einrichtung von W-Seminaren zum Thema „kosmische Strahlung“ soll an den Gymnasien in Erlangen genutzt werden, um SchülerInnen für die Astroteilchenphysik zu begeistern und somit gleichzeitig für ein Studium der Physik und das „Erlangen Centre for Astroparticle Physics“ (**ECAP**) zu werben. Die dieser Arbeit zugrunde liegende Aufgabenstellungen lautet daher:

„Konzeption eines wissenschaftspropädeutischen Seminars zur kosmischen Strahlung“

Zunächst wird in Kapitel 2 ein schultaugliches Detektorkonzept zur Messung von Myonen der kosmischen Strahlung vorgestellt. Kapitel 3 befasst sich mit Aufbau und Ergebnissen erster Versuche mit den Detektoren. Zur Datennahme und Auswertung mussten einige Computerprogramme entwickelt werden, welche in Kapitel 4 ausführlich erläutert werden. Ein langfristiges Ziel des Projekts ist die Einrichtung eines Detektorarrays an Schulen in Erlangen. Erste Abschätzungen der zu erwartenden Ereignisraten sind in Kapitel 5 festgehalten. Um den Gymnasien den Einstieg in dieses bisher „schul-

1 Einleitung und Ziele

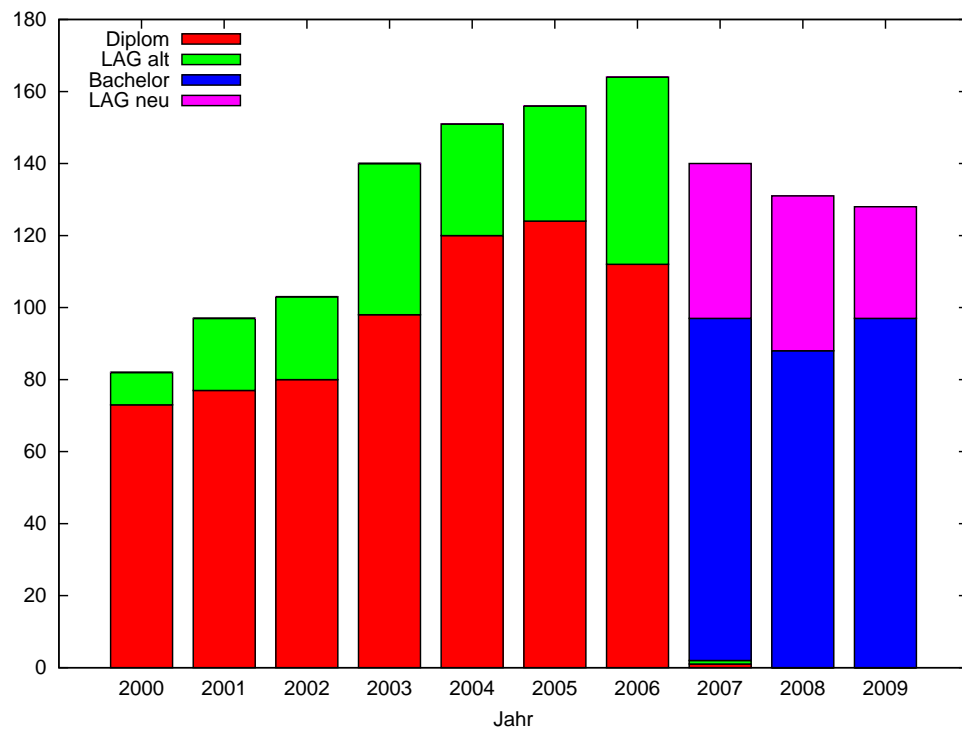


Abbildung 1.1: Zahl der Studienanfänger im Fach Physik an der FAU-Erlangen-Nürnberg [Fri09]

fremde“ Thema zu erleichtern, liefert Kapitel 6 einen Vorschlag zur Konzeption eines W-Seminars zum Thema kosmische Strahlung. Details der Organisation und Partnerschaft mit der Universität Erlangen-Nürnberg können Kapitel 7 entnommen werden. Abgerundet wird diese Arbeit durch ein Fazit mit Ausblick in Kapitel 8.

2 Der Cosmic Ray Muon Detector

Dieses Kapitel beschäftigt sich mit der Hard- und Software, welche im Rahmen dieser Arbeit zur Detektion kosmischer Strahlung verwendet wird. Abschnitt 2.1 widmet sich dem Zusammenbau des Detektors. Auf die Funktion der einzelnen Bauteile wird näher in Abschnitt 2.2 eingegangen. Vor der Inbetriebnahme muss die Installation der benötigten Treiber und Software durchgeführt werden. Dies wird in Abschnitt 2.3 erläutert. Eine Anleitung zur Kalibration der Detektoren wird in Abschnitt 2.4 gegeben.

2.1 Aufbau

Das Herz des Detektors ist das **QuarkNet** „Data-Acquisition-Board“ (**DAQ-Board**), welches in Zusammenarbeit des Fermilab, der Universität von Nebraska und der Universität von Washington entwickelt wurde. Der „Cosmic Ray Muon Detector“ (**CRMD**) wurde geplant, um Schulprojekten die Möglichkeit zu geben, Experimente mit hoch-energetischer kosmischer Strahlung durchzuführen. Der Detektor kann am FermiLab in den USA bestellt werden und wird als Bausatz geliefert. Das Paket besteht aus:

- Vier Szintillatorplatten
- Vier Sekundärelektronenvervielfachern
- Vier Signalkabeln
- Vier Audioverlängerungskabeln
- Netzteil
- Spannungsverteilungsbox
- GPS-Modul
- DAQ-Board
- Hilfsmitteln zum Zusammenbau

2 Der Cosmic Ray Muon Detector

Die Szintillatorplatten werden von der Firma „Eljen Technology“ hergestellt. Im CRMD wird ein Plastik-Szintillator vom Typ „EJ-200“ auf Basis von Polyvinyltoluol verwendet. Die Signalanstiegszeit beträgt 0,9 ns bei einer Abklingdauer von 2,1 ns. Die Szintillations-Effizienz des Materials wird mit 10000 Photonen für ein Elektron mit $E = 1$ MeV angegeben. Weitere Daten sind im Datenblatt des Herstellers [Elj] verfügbar.

Die Sekundärelektronenvervielfacher stammen von der Firma „Sens-Tech“ und sind vom Typ „P30CW5“. Die Bialkali-Photokathode hat ihren sensitiven Bereich im Blau-Grünen. Das Eintrittsfenster hat einen Durchmesser von 30 mm. Dank der verwendeten „Cockroft-Walton-Technologie“ zur Erzeugung der benötigten Hochspannung wird die „Photo-Multiplier-Tube“ (**PMT**) lediglich bei einer Eingangs-Spannung von 5 V betrieben. Genauere Informationen sind der Homepage des Herstellers [Sena],[Senb] zu entnehmen.



(a) Einwickeln in Alu-Folie

(b) Freilegen des Cookie

Abbildung 2.1: Verpacken der Szintillatorplatte

Der Zusammenbau des Detektors erfordert Geduld und einen sehr behutsamen Umgang mit sämtlichen Komponenten. Sollen die Detektoren mit Schülern aufgebaut werden, sollten diese eindringlich darauf hingewiesen werden. Die neuralgischen Punkte sind vor allem die Szintillatorplatten, das DAQ-Board und die Photomultiplier, weil diese Teile eine sehr vorsichtige Behandlung erfordern und zudem sehr teuer sind. Außerdem ist es ratsam, die Arbeit nicht mit mehr als vier Schülern gleichzeitig durchzuführen. Dadurch ist jeder beschäftigt und Langeweile, die zu Unachtsamkeit und schlampigem Arbeiten führt, wird somit verhindert. Immerhin besteht ein Satz aus vier Detektoren, wodurch also 16 Schüler aktiv (wenn auch zeitlich versetzt) an der Konstruktion mitwirken können. Für das Zusammensetzen der Detektoren werden folgende Werkzeuge benötigt:

2 Der Cosmic Ray Muon Detector

- Schere
- Skalpell
- Baumwollhandschuhe
- Klebeband
- Teflonband
- lichtundurchlässiges Isolierband

Die mitgelieferte Alu-Folie wird mit einer Schere in vier gleich große Rechtecke zerteilt. Es handelt sich hierbei um eine Spezialfolie, welche etwas dicker ist als die handelsübliche Aluminiumfolie. Anschließend wird eines der Stücke auf eine Größe von $48\text{ cm} \times 68\text{ cm}$ zugeschnitten. Die Folie wird mit der matten Seite nach unten auf einen Tisch ohne Unebenheiten gelegt und die Falten werden durch vorsichtiges Streifen mit den Händen (Handschuhe!) so gut als möglich ausgebügelt. Geschieht dies mit zuviel Druck oder ohne Handschuhe, kann die Folie reißen.



Abbildung 2.2: Lichtdicht verpackte Szintillatorplatte

Im nächsten Schritt wird die Szintillatorplatte auf die glänzende Seite der Folie so gelegt, dass die Folie anschließend über die langen Kanten der Platte gefaltet werden kann. Auch wenn die Szintillatorplatte nicht sauber sein sollte, sollte keinesfalls versucht werden, diese zu reinigen. Das Szintillatormaterial ist sehr empfindlich und in der Regel wird das Ergebnis nach dem Putzen eher schlechter als besser aussehen. Die Platte wird nun wie ein Geschenk eingepackt und die Folie mit Klebeband fixiert (siehe Abbildung 2.1(a)). Der „Cookie“ muss nun mit einer Rasierklinge oder einem Skalpell freigelegt werden (siehe Abbildung 2.1(b)). Dabei gilt es, mit äußerster Vorsicht zu agieren und keinesfalls Platte oder Cookie mit dem Schneidewerkzeug zu treffen. Beim Falten muss auf die scharfen Kanten der Szintillatorplatten geachtet werden, weil dadurch leicht Risse in der Folie entstehen können. Sollten nach dem Einpacken Löcher

oder Risse entdeckt werden, sind diese mit den vorher abgeschnittenen Folienresten zu überkleben, so dass die Platte vollständig verpackt ist. Jetzt wird die Mantelfläche



(a) Auftragen des optischen Fetts



(b) Verbindung von Szintillator und PMT

Abbildung 2.3: Anbringung des PMT an den Szintillator

des Cookie sorgfältig mit Teflonband umwickelt. Die einzige Sichtmöglichkeit in die Szintillatorplatte sollte die Kreisfläche des Cookie sein. Im nächsten Schritt muss die schwarze, lichtundurchlässige Folie in vier Teile der Größe $25\text{ cm} \times 60\text{ cm}$ zerschnitten werden. Ein Teil wird mit der schwarzen Seite nach unten zeigend auf den Tisch gelegt. Die Platte wird so auf die Folie gelegt, dass diese anschließend über die kurze Kante ohne Cookie gefaltet werden kann. Mit etwas Isolierband wird die schwarze Folie straff fixiert. Die Kanten werden mit dem breiten Isolierband überklebt. Zuletzt sollte die Region um den Cookie noch mit schmalen Isolierband überdeckt werden (siehe Abbildung 2.2).

Im darauffolgenden Schritt wird der PMT benötigt. Die Schutzkappe wird abgezogen und etwas optisches Fett auf das Glasfenster aufgetragen (siehe Abbildung 2.3(a)). Anschließend wird der PMT mit leichtem Druck an den Cookie gepresst und Szintillatorplatte und PMT durch Umwickeln mit Isolierband verbunden (siehe Abbildung 2.3(b)). Dabei ist besonders auf das empfindliche Eintrittsfenster des PMT zu achten.

Im letzten Schritt werden mit äußerster Vorsicht PMT und Platte in das Plastikrohr zur Stabilisierung eingeführt. Dabei ist zum einen auf die Kabel des PMT zu achten und zum anderen auf den empfindlichen PMT. Sitzt das Gespann weit genug im Stabilisator, wird die Schutzkappe des PMT bis zum Anschlag an die Szintillatorplatte eingeführt und das Ende des Rohrs mit Isolierband umwickelt. Der Spalt zwischen PMT und Szintillatorplatte wird ebenfalls mit Isolierband verschlossen (siehe Abbildung 2.4). Auch das Kabelende des Rohrs sollte mit Isolierband verschlossen werden. Im praktischen Betrieb hat es sich als nützlich erwiesen, auch den Übergang von Szin-

2 *Der Cosmic Ray Muon Detector*



Abbildung 2.4: Fertiggestelltes Detektor-„Paddel“

tillatorplatte und Stabilisatorrohr mit schwarzem Klebeband zu überdecken. Damit ist Detektor fertiggestellt und kann kalibriert werden.

2.2 Funktion

Sämtliche Informationen über die Funktionsweise des CRMD wurden dem Handbuch [RJPB10] entnommen. Das CRMD besteht aus:

- Vier Szintillator-Paddeln mit PMT-Auslese (siehe Abschnitt 2.1)
- DAQ-Board
- GPS-Modul und -Antenne
- Netzteil zur Spannungsversorgung
- Spannungsregelungsmodul für die PMTs

Das Herzstück des Systems ist das DAQ-Board (siehe Abbildung 2.5). Es besitzt vier Signaleingänge, einen Anschluß für das GPS-Modul, eine USB-Schnittstelle, einen Spannungseingang und einen Spannungsausgang für das Spannungsregelungsmodul der vier Photomultiplier.

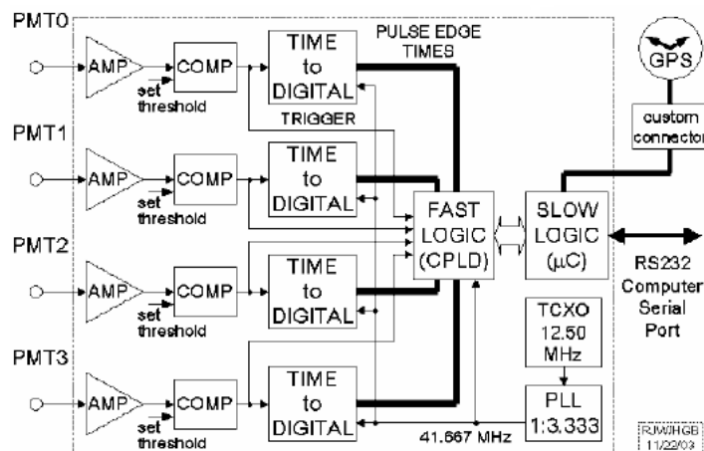


Abbildung 2.5: Schaltbild des DAQ-Boards [RJPB10]

2.2.1 Input

Spannungsquelle

Das Board wird mit einem 5 V-Gleichspannungsnetzteil betrieben (im Lieferumfang enthalten). Jedoch wird nur ein Kabel mit Stecker-Typ A mitgeliefert, der zwar in den

USA üblich ist, aber in Deutschland nicht verwendet wird. Für den Einsatz in Deutschland muss ein Kabel mit Stecker-Typ CEE (gängig in Europa) und Kleingerätestecker IEC-60320-C7 beschafft werden.

Steuerung

Das DAQ-Board kann über die USB-Schnittstelle gesteuert werden. Es wird ein Terminal-Programm (siehe Abschnitt 2.3.2 und Abschnitt 4.2.1) benötigt, um die Steuerkommandos einzugeben und an das DAQ-Board zu senden. Die wichtigsten Steuerbefehle können Tabelle 2.1 entnommen werden. Alle Befehle und deren genaue Beschreibung sind in Abschnitt A aufgelistet.

Tabelle 2.1: Wichtige Steuerbefehle

Befehl	Bedeutung	
CE	Counters Enable	Start der Datenaufzeichnung
CD	Counters Disable	Ende der Datenaufzeichnung
DG	Display GPS	Anzeigen der GPS-Informationen
DS	Display Scaler	Anzeigen der Zählerstände
RB	Reset Counters	Zurücksetzen der Zähler
RE	Reset Everything	Zurücksetzen aller Einstellungen
SA	Save	Speichern der Einstellungen
ST	Status	Ausgabe einer Statuszeile
TL	Threshold Level	Eingabe der Schwellenspannungen
WC	Write Control Registers	Setzen der Koinzidenzparameter
WT	Write TMC Registers	Setzen der Koinzidenzparameter

Zeit-, Positions-, Luftdruck- und Temperaturinformation

Die Zeit-, Positions- und Temperaturinformation wird von einem GPS-Modul geliefert, welches durch ein CAT5-Kabel mit dem DAQ-Board verbunden wird. An das Modul muss die wetterbeständige GPS-Antenne angeschlossen und mit freier Sicht auf eine Himmelshälfte ausgerichtet werden. Laut Handbuch [RJPB10] behindern Fenster die Sicht der Antenne, was jedoch in den Tests nicht bestätigt werden konnte. Das Barometer zur Messung des Luftdrucks befindet sich auf dem DAQ-Board. Vor der Verwendung muss es mit Hilfe eines zweiten Barometers kalibriert werden. Der Temperatursensor ist für den Außeneinsatz geeignet und muss mit dem GPS-Modul verbunden werden.

Signal

An den vier Eingängen können BNC-Signalkabel angebracht werden. Die empfangenen Signale werden dann vom DAQ-Board im „Amplifier“ (**AMP** in Abbildung 2.5) um den Faktor zehn verstärkt. Anschließend durchläuft das skalierte Signal den „Discriminator“ (**COMP** in Abbildung 2.5), der entscheidet, ob das Signal aufgezeichnet wird. Dazu muss die Amplitude des Signals eine bestimmte Schwelle, den „Threshold-Level“ (**TL**), übersteigen. Diese Schwelle kann vom Benutzer auf einen Wert zwischen 0 mV und 600 mV gesetzt werden. Um wenig Rauschen und möglichst viele „echte Signale“ zu erhalten, muss der optimale TL-Wert bei der Kalibration (siehe Abschnitt 2.4) ermittelt werden.

2.2.2 Timing

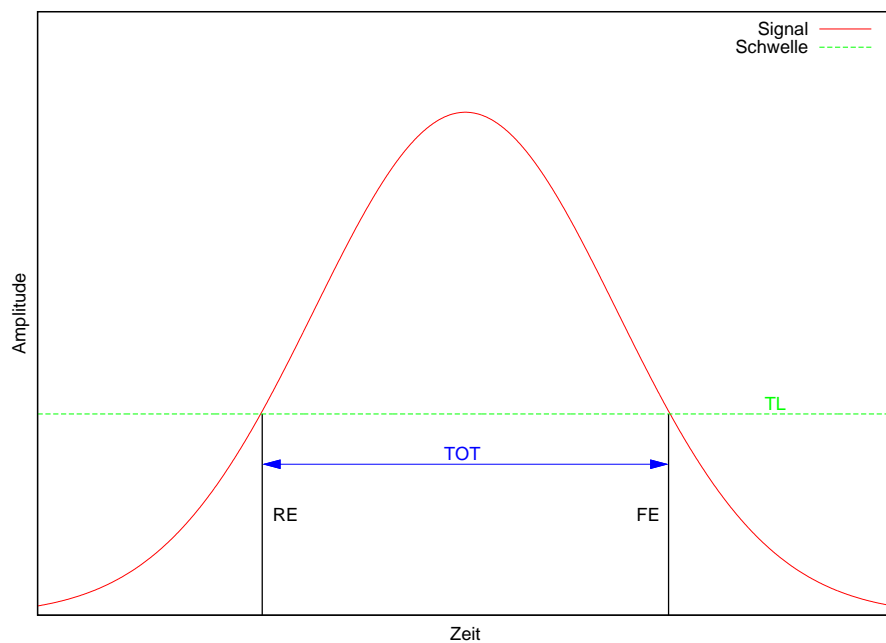


Abbildung 2.6: Zusammenhang von TL, RE, FE, TOT (GnuPlot)

Wurde ein Signal vom COMP-Modul zugelassen, so wird ihm vom „Time-to-Digital-Converter“ (**TDC**) die Zeitinformation zugewiesen. Diese besteht aus fünf Komponenten:

1. GPS-Zeit
2. Verzögerung des GPS-Signals
3. Zählerstand des 25 MHz-Chips beim letzten 1pps-Signal

4. aktueller Zählerstand des 25 MHz-Chips
5. RE/FE-Zeitinformation

Die GPS-Zeit wird bis auf ms angegeben. Die Verzögerung des 1pps-Signals wird auch in ms angegeben. Um nun genauere Zeitinformationen bis in den ns –Bereich zu erhalten, werden der aktuelle Zählerstand und der Zählerstand beim letzten 1pps-Signal des 25 MHz-Zeitmessungschips benötigt. So wird die Grundzeit eines Ereignisses bis auf 40 ns erhalten. Mit „Rising Edge“ (**RE**) und der „Falling Edge“ (**FE**) (siehe Abbildung 2.2.2), welche der „Time-Measurement-Chip“ (**TMC**) in einer Auflösung von 1,25 ns (40 ns/32) zurückgibt, ist die Zeitinformation für ein Ereignis komplett. Die Zeitspanne zwischen RE und FE ist die „Time-over-Threshold“ (**TOT**) , während welcher sich das Signal oberhalb der Schwelle befindet.

2.2.3 Koinzidenzbedingungen

Das „Complex-Programmable-Logic-Device“ (**CPLD**) überprüft, ob die eintreffenden Signale die Koinzidenzkriterien erfüllen und somit aufgezeichnet werden sollen. Dazu wird dem CPLD die Anzahl der Kanäle übergeben, die bei einem Ereignis koinzident getroffen werden müssen. Dies geschieht mit dem Board-Befehl `WC 00 ab` . $a \in \{0; 1; 2; 3\}$ gibt die Anzahl der gleichzeitig zu treffenden Kanäle weniger eins an. b ist eine hexadezimale Ziffer und steht für die Auswahl der Kanäle. Dabei repräsentiert b den Nibble (4 Bits) ($B_{CH0} B_{CH1} B_{CH2} B_{CH3}$), wobei $B_{CHx} \in \{0; 1\}$ den Wert 1 annimmt, falls der Kanal x aktiviert ist. Soll das Set zum Beispiel in Zwei-Kanal-Koinzidenz ($a = 1$) betrieben werden und der Kanal 0 ausgeschlossen werden ($b = (7)_{16} = (0111)_2$), so lautet der Eingabebefehl `WC 00 17` .

Außerdem muss über die Parameter Trigger-Zeit und Gate-Breite die Koinzidenz definiert werden. Beide Parameter werden in 40-ns-Schritten übergeben. Die Trigger-Zeit ist das Zeitintervall, in welchem die Anzahl der vorher gewählten Kanäle getroffen werden muss, um als Ereignis erkannt und aufgezeichnet zu werden. Sie wird durch die Board-Befehle `WT 01 00` und `WT 02 ab` festgelegt. ab ist eine zweistellige Hexadezimale und gibt die Dauer der Trigger-Zeit an. Der maximale Wert für ab liegt bei $(7F)_{16} = (127)_{10}$, was einer Zeit von 5080 ns entspricht. Ist beispielsweise eine Trigger-Zeit von 400 ns gewünscht, so ergibt sich $ab = (0A)_{16} = (10)_{10} = \left(\frac{400 \text{ ns}}{40 \text{ ns}}\right)_{10}$. Die Befehle müssen somit lauten: `WT 01 00` und `WT 02 0A` .

Die Dauer der Aufzeichnung, die durch den Trigger gestartet wird, wird durch die

Gate-Breite festgelegt. Sie wird dem DAQ-Board mit den Befehlen `WC 02 cd` und `WC 03 ab` mitgeteilt. Ist x also die Gate-Breite in 40-ns-Ticks, dann gilt $(x)_{10} = (abcd)_{16}$. Soll der Wert des Parameters $10\ \mu\text{s}$ betragen, so ergibt sich $(abcd)_{16} = (00FA)_{16} = (250)_{10} = \left(\frac{10000\ \text{ns}}{40\ \text{ns}}\right)_{10}$. Somit müssen die Befehle `WC 02 FA` und `WC 03 00` lauten.

2.2.4 Output

Die Daten werden über die USB-Schnittstelle unter Verwendung einer Terminal-Software (siehe Abschnitt 2.3.2) ausgegeben. Die Schnittstelle wird dabei von einer „Micro-Controller-Unit“ (**MCU**) betrieben. Die Daten erscheinen im Hex-Format:

```
3774BBFE 80 00 00 00 00 00 00 3C 00 370A867C 000021.022 030610 A 08 0 +0072
3774BBFF 00 00 23 00 00 00 00 00 00 370A867C 000021.022 030610 A 08 0 +0072
3774BBFF 00 00 00 31 00 00 00 00 00 370A867C 000021.022 030610 A 08 0 +0072
3774BC00 00 00 00 00 00 00 00 00 21 370A867C 000021.022 030610 A 08 0 +0072
```

Eine Zeile besteht aus 16 Wörtern. Die Bedeutung der Wörter kann Tabelle 2.2 entnommen werden. Für genauere Informationen ist das Handbuch [RJPB10] zu Rate zu ziehen.

Aus jeder Datenzeile kann direkt die Zeit $T = T_G + T_K$ eines Ereignisses ermittelt werden. Sie setzt sich zusammen aus der Grundzeit T_G , welche mit Gleichung (2.1) berechnet wird, und der Kanalzeit T_K , die aus dem RE/FE des jeweiligen Kanals hervorgeht.

$$T_G = \lfloor \text{Wort 11 in s} + \frac{\text{Wort 16 in ms}}{1000} \rfloor + \frac{\text{Wort 1} - \text{Wort 10}}{f_{\text{clock}}} \quad (2.1)$$

Dabei ist $\lfloor x \rfloor$ die „Floor-Funktion“ (Abrundung auf Ganzzahl). Die beiden Summanden innerhalb der Gaußklammer in Gleichung (2.1) liefern die Sekunden seit Mitternacht bis zum Zeitpunkt eines Ereignisses, indem zur GPS-Zeit die Verzögerungskorrektur addiert wird und die Summe anschließend auf Sekunden gerundet wird. Aus dem dritten Summanden in Gleichung (2.1) errechnet sich die hochauflösende Zeitinformation. Dabei wird der Zählerstand der CPU bei Eintreffen des letzten 1pps-Signals vom aktuellen Zählerstand abgezogen und durch die Frequenz der CPU $f_{\text{clock}} = 25\ \text{MHz}$ geteilt. Die Kanalzeiten T_K werden aus den Wörtern 2 - 9 berechnet, indem der entsprechende Zähler mit $\frac{40\ \text{ns}}{32}$ multipliziert wird.

2 Der Cosmic Ray Muon Detector

Tabelle 2.2: Bedeutung der Rohdaten

Wort	Format	Einheit	Bedeutung
1	hex	40 ns/count	Zähler der 25 MHz-DAQ-Uhr
2	hex	1,25 ns/count	RE Kanal 0
3	hex	1,25 ns/count	FE Kanal 0
4	hex	1,25 ns/count	RE Kanal 1
5	hex	1,25 ns/count	FE Kanal 1
6	hex	1,25 ns/count	RE Kanal 2
7	hex	1,25 ns/count	FE Kanal 2
8	hex	1,25 ns/count	RE Kanal 3
9	hex	1,25 ns/count	FE Kanal 3
10	hex	40 ns/count	Zählerstand beim letzten 1pps-Signal
11	dec	HHMMSS.mmm	GPS-Zeit
12	dec	ddmmyy	Datum
13	-	A valid, V invalid	Status der GPS-Information
14	dec	-	Anzahl verfügbarer GPS-Satelliten
15	hex	-	Status des DAQ-Boards
16	dec	1 ms	Verzögerung des 1pps-Signals

2.3 Installation von Treibern und Software

Das Auslesen des DAQ-Board setzt die Installation eines Treibers für den Board-Chip voraus. Um die Daten auf dem Computer empfangen und speichern zu können, wird eine Terminal-Software benötigt. Vor Beginn der Datennahme müssen noch die entsprechenden Einstellungen für eine reibungslose Kommunikation zwischen Board und PC getroffen werden.

2.3.1 Treiber für USB-Schnittstelle

Um das DAQ-Board über den PC auslesen und steuern zu können, ist ein Treiber für den auf dem Board verwendeten **Silabs CP210x**-Chip notwendig. Dieser kann für verschiedene Betriebssysteme auf der Homepage des Herstellers [Sil] heruntergeladen werden. Bei der Installation ist den Bildschirmanweisungen Folge zu leisten.

2.3.2 Terminal-Software

Zur Kommunikation mit dem DAQ-Board wird eine Terminal-Software benötigt. Wird ein **Windows**-Betriebssystem verwendet, ist das **Hyperterminal** schon im Paket enthalten. Es befindet sich im Ordner „C:\Programme\Windows NT\hypertrm.exe“ oder kann über den Start-Button unter „Alle Programme -> Zubehör -> Kommunikation -> Hyperterminal“ gestartet werden. Sollte das Programm nicht installiert sein, kann es über „Start -> Systemsteuerung -> Software -> Windows Komponenten -> Zubehör und Dienstprogramme -> Details -> Kommunikation -> Details -> Hyperterminal“ aktiviert werden.

Es können auch andere Terminal-Programme (z.B. **Zterm** für **Windows** und **Mac**, **screen** unter **linux**) verwendet werden. Im Rahmen dieser Arbeit wurde auch ein eigenes Terminalprogramm (**getdata** siehe Abschnitt 4.2.1) entwickelt, da bei der Verwendung des Windows Hyperterminal öfters Probleme auftraten, die zu großen Datenverlusten führten. Unabhängig davon welches Terminal benutzt wird, wird von **QuarkNet** empfohlen mit dem Board-Befehl **ST 2 5** die Statuszeile zu aktivieren, welche dann alle fünf Minuten in den Daten erscheint.

2.3.3 Einstellung des Terminals

Das Terminal benötigt den COM-Port **X**, welcher vom DAQ-Board verwendet wird. Dieser ist in Windows zu finden unter: „Systemsteuerung -> System -> Hardware -> Geräte-Manager -> Anschlüsse (COM und LPT) -> Silicon Labs CP210x USB to UART Bridge (COM**X**)“. Außerdem müssen der Terminalsoftware fünf Parameter übergeben werden, um eine korrekte Verbindung mit dem DAQ-Board herzustellen. Die korrekten Werte für die Parameter sind in Tabelle 2.3 aufgeführt. Im Terminalprogramm **getdata** sind die richtigen Parameter schon voreingestellt (siehe Quelltext in B.2 Zeile 84/85). Treten Probleme bei der Datenübertragung auf, hilft meist ein Beenden der Verbindung und ein anschließender Neustart des Terminals.

Tabelle 2.3: Parameter zur Einstellung des Terminals

Parameter	Wert
Baud-rate	115200
Data bits	8
Parity	None
Stop bits	1
Flow control	Xon/Xoff

2.4 Kalibration

Eine sorgfältige Kalibration ist die Basis für zuverlässige Daten. Dabei sind für jeden Detektor die beiden Parameter Schwellenspannung (**Threshold Level**) der COMP-Module und Spannungsversorgung der PMTs so zu bestimmen, dass der Detektor zum einen möglichst wenig Untergrundrauschen, zum anderen möglichst viele „echte“ Signale aufzeichnet.

2.4.1 Bestimmung der Schwellenspannung

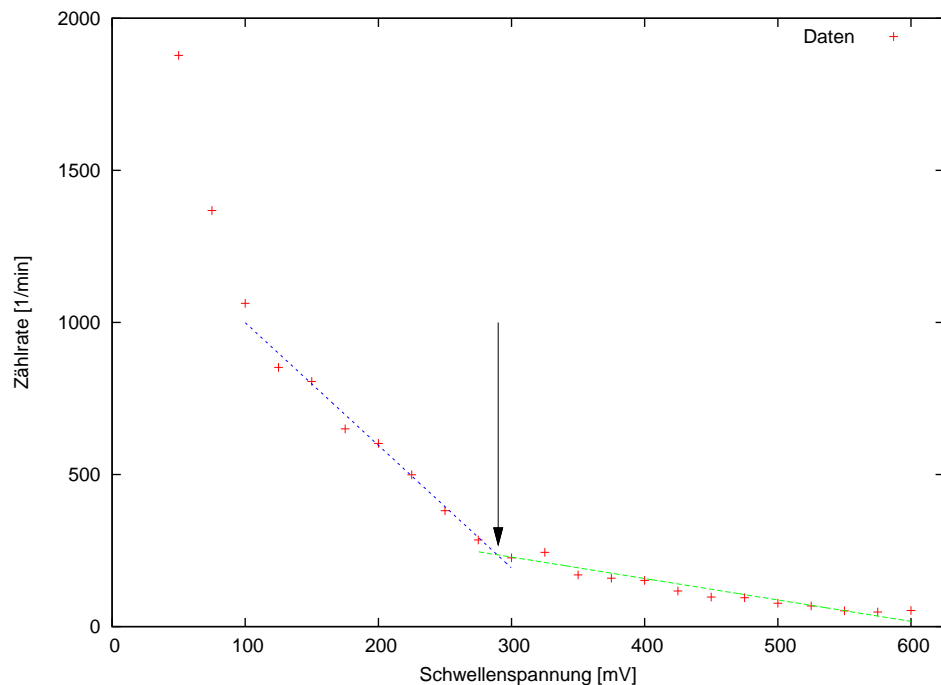


Abbildung 2.7: Messung der der Zählrate in Abhängigkeit der Schwellenspannung für einen Kanal (GnuPlot)

Die Schwellenspannung wird über das DAQ-Board gesteuert und kann vom Benutzer für alle vier Kanäle eingestellt werden. Zur Kalibration der Schwellenspannung wird die Spannungsversorgung der Detektoren konstant gehalten und dabei die Zählrate für verschiedene Schwellenspannungen aufgenommen. Zuerst wird der Wert der Versorgungsspannung ermittelt, ab welchem der Detektor Ereignisse aufzeichnet. Dazu wird die Schwelle TL auf 600 mV gesetzt und die Zähler aktiviert. Anschließend wird die Versorgungsspannung solange erhöht, bis das Terminal anfängt, Datenreihen aufzuzeichnen. Anschließend werden die Raten für Schwellen von 25 mV bis 600 mV bei den eben ermittelten Werten der Spannungsversorgung eine bestimmte Zeit lang (ca. 5 min) aufgenommen. Werden nun die Zählraten über den Schwellenspannungen

aufgetragen, kann die optimale Schwellenspannung, als der Wert, ab dem die Zählrate annähernd linear wird, abgelesen werden (siehe Abbildung 2.7). Mit Hilfe des java-Programms **autocal** (siehe Abschnitt 4.1) entfällt das manuelle Schwellensetzen und anschließende Abwarten. Das Zeitintervall kann beim Programmaufruf übergeben werden. Nach Durchlaufen des Programms können die Daten mit einem Tabellenkalkulationsprogramm weiterverarbeitet und daraus die Schwellen ermittelt werden.

2.4.2 Bestimmung der Spannungsversorgung im Single-Modus

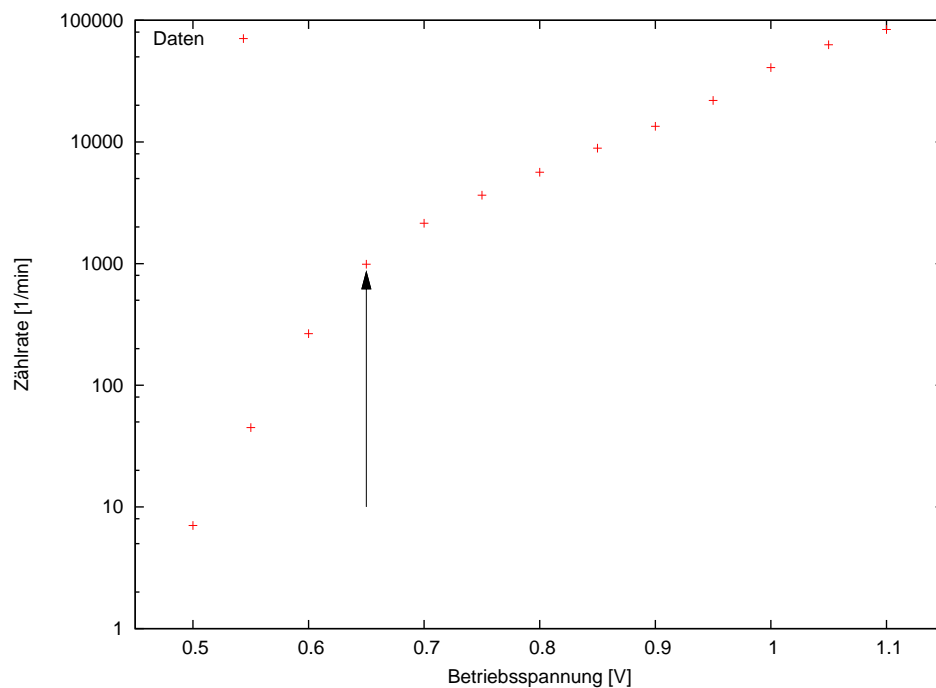


Abbildung 2.8: Messung der Zählrate in Abhängigkeit der Betriebsspannung im Single-Modus für einen Kanal (GnuPlot)

Im nächsten Schritt ist es wichtig, die optimale Betriebsspannung für die PMT zu finden. Dies geschieht zunächst im Single-Modus, wobei für jeden Detektor einzeln die Zählrate in Abhängigkeit von der Betriebsspannung bestimmt wird. Hierzu werden die in Abschnitt 2.4.1 ermittelten Schwellenspannungs-Werte auf dem DAQ-Board eingestellt. Anschließend werden für alle Detektoren die Zählraten über einen gewissen Zeitraum (ca. 5 min) für Betriebsspannungen zwischen 0,3 V und 1,2 V aufgenommen. Werden nun die Zählraten logarithmisch über der Spannung aufgetragen, kann eine Abflachung der Steigung des Graphen ab einem bestimmten Wert erkannt werden (siehe Abbildung 2.8). Dieser Wert ist die ideale Betriebsspannung für den entsprechenden Detektor. Der Nachteil dieser Methode liegt jedoch darin, dass die genau Lage des Knicks meist nur schwer abzulesen ist. Die Zählrate ist proportional zur Betriebsspan-

nung der PMT. Wird sie falsch eingestellt, wirkt sich das direkt auf die Messungen aus.

2.4.3 Bestimmung der Spannungsversorgung im Koinzidenz-Modus

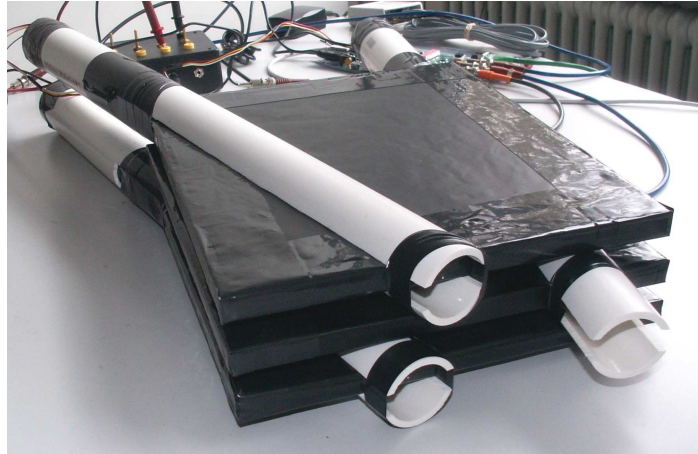


Abbildung 2.9: Anordnung zur Kalibration im Koinzidenz-Modus

Zur genaueren Ermittlung der korrekten Spannungsversorgung wird die Koinzidenz-Methode benutzt. Dabei werden die Detektoren wie ein „Sandwich“ aufgeschichtet (siehe Abbildung 2.9). Nach Einstellung der ermittelten Schwellenspannungen wird zuerst der Referenzwert bestimmt. Da angenommen werden kann, dass der Teilchenfluss durch die Detektoren konstant bleibt, werden zwei Detektoren in Zweier-Koinzidenz geschaltet und die Koinzidenzrate gemessen. Anschließend wird der zu kalibrierende Detektor mit einem der Referenzdetektoren in Koinzidenz geschaltet und die Koinzidenzrate aufgenommen, wobei lediglich die Betriebsspannung des zu kalibrierenden Detektors variiert wird. Wird nun das Verhältnis der Koinzidenzraten des zu kalibrierenden Detektors zum Referenzwert logarithmisch aufgetragen, ist deutlich ein Plateau zu erkennen (siehe Abbildung 2.10). Die optimale Betriebsspannung lässt sich nun als der Wert ablesen, bei welchem der abgeflachte Teil beginnt.

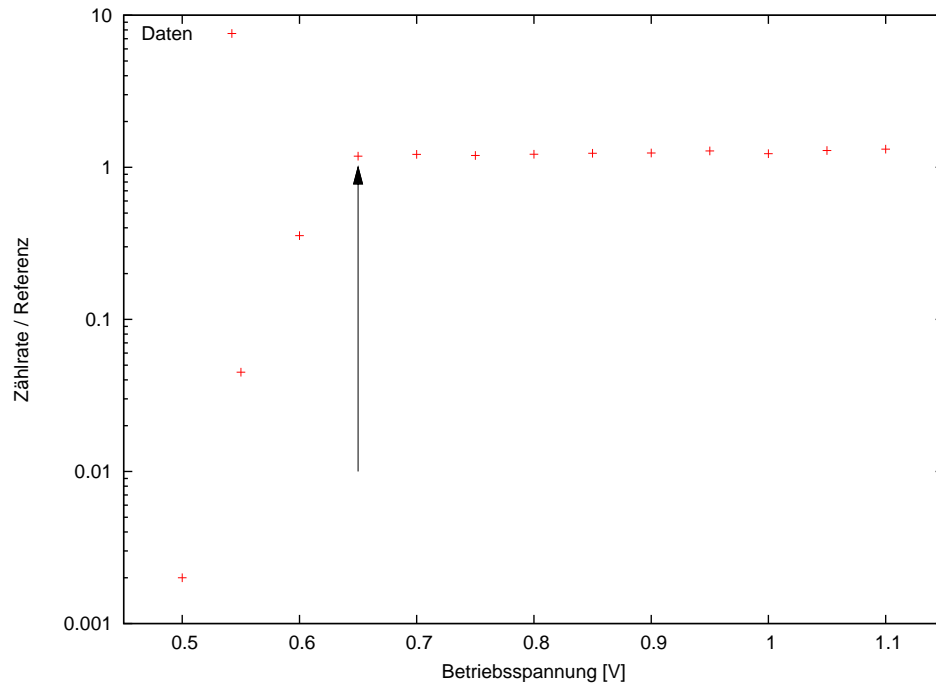


Abbildung 2.10: Messung der Zählrate in Abhängigkeit der Betriebsspannung im Koinzidenz-Modus für einen Kanal (GnuPlot)

2.4.4 Optimierung

Da sich die Charakteristiken der PMTs mit der Zeit verändern, müssen die Detektoren öfters nachkalibriert werden. Dabei ergab sich folgende Routine zur Kalibration, um iterativ die optimalen Einstellungen zu finden:

1. TL aller Kanäle auf 500 mV setzen, anschließend Spannung der PMTs erhöhen bis Signale aufgezeichnet werden. Diese Spannungswerte für die Spannungsversorgung verwenden.
2. Optimale TL-Werte für jeden Kanal mit Hilfe von **autocal** ermitteln (siehe Abschnitt 2.4.1).
3. Optimale Werte für Spannungsversorgung mit Koinzidenz-Methode (siehe Abschnitt 2.4.3) bestimmen.
4. Kalibration durch Leistungsanalyse (siehe Abschnitt 3.1) überprüfen. Bei schlechter Kalibration wieder mit Schritt 2 beginnen.

3 Versuche mit dem Cosmic Ray Muon Detector

Ein Bestandteil dieser Arbeit ist das Durchführen von verschiedenen Experimenten mit dem CRMD-Set zur Erprobung seiner Einsetzbarkeit im Schulunterricht. Begonnen wurde mit einem Versuch, der der Überprüfung der Kalibrationsparameter dient (Abschnitt 3.1). Das in Abschnitt 3.2 dargestellte Experiment zur Messung der Myonenrate warf viele technische Probleme auf, die im Verlauf der Messungen beseitigt werden konnten. Abschnitt 3.3 befasst sich mit der Ermittlung der Myonenlebensdauer. In Abschnitt 3.4 wurden die Herkunftsrichtungen ausgedehnter Luftschauer anhand von CRMD-Daten rekonstruiert. Abschließend werden in Abschnitt 3.5 noch Ideen für weitere Versuche mit dem Detektorset aufgezeigt.

3.1 Analyse der Leistungsfähigkeit des Detektorsets

Nach der Kalibration des **CRMD** sollte die Leistungsfähigkeit des Detektorsets überprüft werden. Da das DAQ-Board neben der **TDC**-Einheit leider keine **ADC**-Einheit besitzt, ist keine Energiekalibration möglich. Fliegt ein Teilchen durch den Szintillator, werden die Zeiten des Über- und Unterschreitens der Schwellenspannung gemessen. Daraus kann die „Time-over-Threshold“ (TOT) bestimmt werden (siehe Abbildung 3.1), welche proportional zur Energie des Teilchens ist. Die TOT ist mit den beiden Kalibrationsparametern Schwellenspannung und Spannungsversorgung verknüpft. Abbildung 3.1 zeigt, dass bei Wahl einer niedrigeren Schwelle die TOT zunimmt. Die Amplitude des Signals ist proportional zur Spannungsversorgung. Wird sie erhöht, steigt die TOT ebenfalls an.

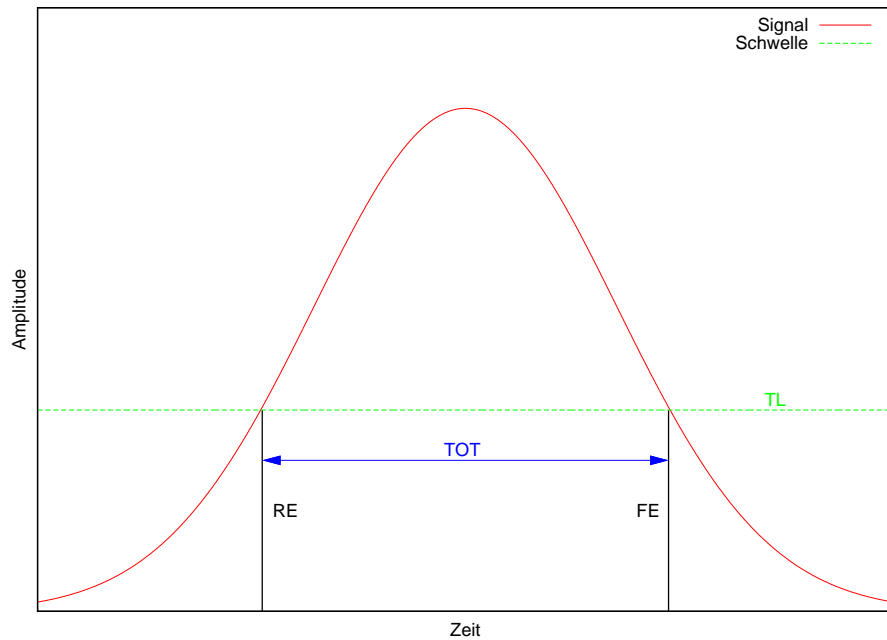


Abbildung 3.1: Beziehung zwischen Signal und **TOT** (GnuPlot)

3.1.1 Aufbau und Methoden

Die Analyse der Leistungsfähigkeit des Detektorsets erfordert keinen speziellen Aufbau. Es empfiehlt sich, die Leistungsanalyse stets nach abgeschlossener Kalibration durchzuführen. Somit kann die Güte der Kalibration überprüft und eventuell falsch gewählte Parameter können korrigiert werden. Das Detektorset ist dabei in Ein-Kanal-Koinzidenz zu betreiben. Dadurch ist die Einstellung der Trigger-Zeit und Gate-Breite unwichtig.

3.1.2 Daten und Analyse

Die Datennahme erfolgt über ein Terminal-Programm (siehe Abschnitt 2.3.2). Zur Analyse der Daten kann das „Cosmic Ray e-Lab“ [Fer] verwendet werden. Dazu müssen die Daten auf die Online-Plattform hochgeladen werden und können anschließend über den Punkt „Performance“ (siehe Abbildung 3.2) verarbeitet werden. Als Ergebnis wird ein Graph erzeugt, der die Anzahl der Counts über der TOT aufträgt. Alternativ kann das Java-Werkzeug gnutot (Abschnitt 4.3.2) zur Auswertung benutzt werden. Die dadurch erhaltenen Daten können mit einem Plotprogramm dargestellt werden.

Der Verlauf des Histogramms der TOT sollte einem Gaußprofil entsprechen. Zeigt

3 Versuche mit dem Cosmic Ray Muon Detector

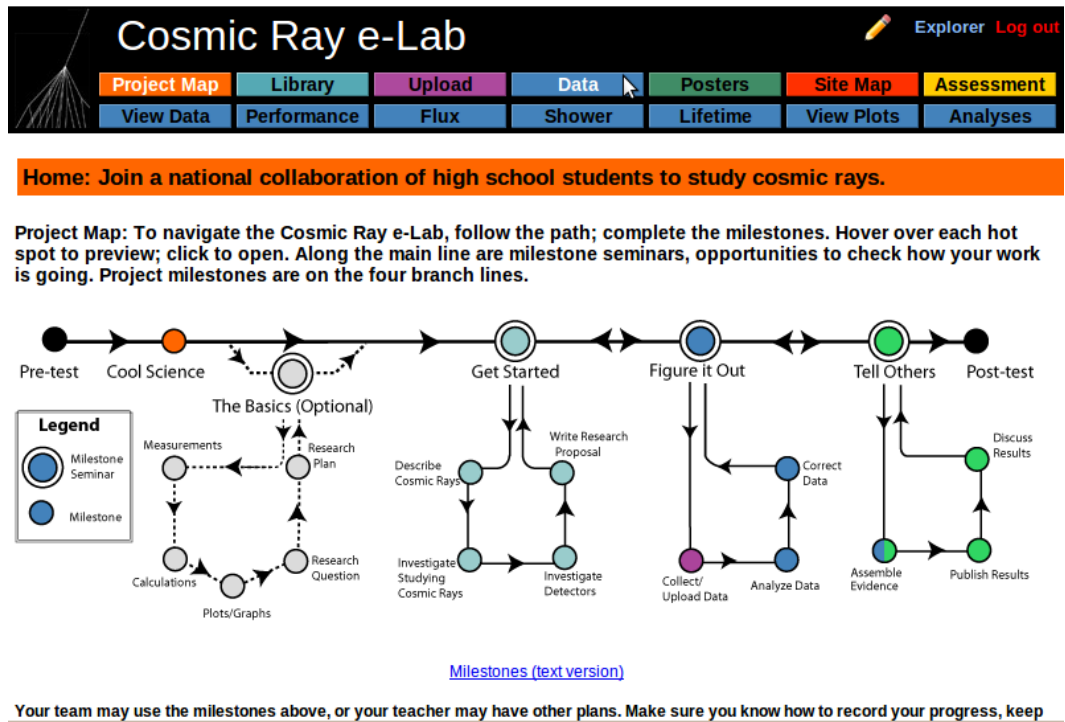


Abbildung 3.2: Startseite der Online-Plattform „Cosmic Ray e-Lab“

das Profil starke Abweichungen davon, sollte die Kalibration nachgebessert werden. Oft ist deutlich ein kleinerer Peak vor dem Hauptpeak oder viele Peaks mit einem unregelmäßigen Verlauf (siehe Abbildung 3.3(a), Daten vom 22.04.2009, Set I, Kanal 4) zu beobachten. Die Ursache dafür liegt im Rauschen des Photomultipliers. Um diese Effekte herauszufiltern, sollte die Schwellenspannung erhöht werden.

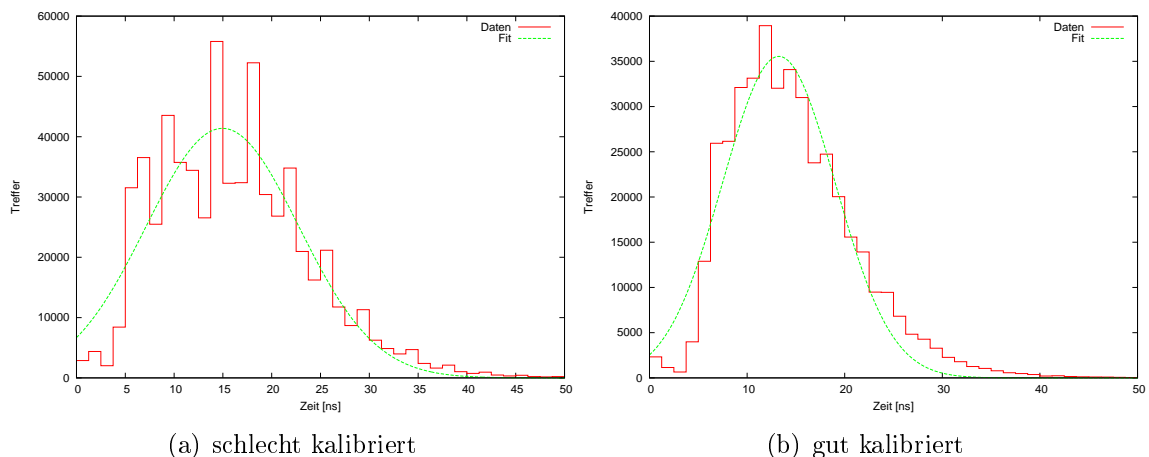


Abbildung 3.3: Analyse der TOT (GnuPlot)

Das in Abbildung 3.3(b) dargestellte TOT-Profil wurde vom Kanal 1 des Detektorsets II am 28.04.2010 aufgezeichnet. Die Auswertung erfolgte mit dem Java-Programm gnutot, erzeugt wurde die Grafik mit GnuPlot. Die Stufenbreite des Histogramms be-

trägt 1,25 ns. Die TOT-Verteilung ähnelt stark dem Gaußprofil mit einem Mittelwert von 18,4 ns. Der Detektor ist somit sehr gut kalibriert und kann mit den eingestellten Parametern für die weiteren Messungen verwendet werden.

3.1.3 Fehler

Bei der Erstellung der Histogramme muss beachtet werden, dass die Zeitauflösung 1,25 ns beträgt. Somit ist es sinnlos, eine kleinere Schrittweite zu wählen. Es ist möglich, dass sich während des Dauerbetriebs die Charakteristiken der PMTs verändern. Dies kann durch eine Leistungsanalyse des Detektors nachgewiesen werden. Nimmt bei gleicher Messdauer die Anzahl der Counts zu, so muss die Spannungsversorgung verringert werden, um dies auszugleichen. Verlagert sich die mittlere TOT hin zu größeren Werten, muss der Threshold-Level erhöht werden, um zum Ausgangszustand zurückzukehren.

3.2 Messung des Myonenflusses

Die Messung des Myonenflusses war eines der ersten durchgeführten Experimente mit dem CRMD im Rahmen dieser Arbeit. Dabei soll der zeitliche Verlauf der Myonenrate bestimmt werden. Schwankungen in der Rate sollten sich auf natürliche Einflüsse (z.B. Luftdruckänderungen) zurückführen lassen. Zu Beginn des Versuchs wurden sehr starke Schwankungen, Sprünge und sogar komplett fehlende Daten registriert, die sich nicht durch natürliche Einflüsse erklären ließen, sondern auf Probleme der verwendeten Soft- und Hardware hindeuteten. Diese Probleme konnten im weiteren Verlauf beseitigt werden, so dass nun eine nahezu störungsfreie Datennahme möglich ist.

3.2.1 Aufbau und Methoden



Abbildung 3.4: Möglicher Aufbau zur Myonenfluss-Messung

Zur Messung des Myonenflusses empfiehlt sich eine Koinzidenzmessung mit einem vertikal gestapelten Aufbau der Detektoren (siehe Abbildung 3.4). Über den Abstand der einzelnen Detektoren kann die Größe des betrachteten Raumwinkels geändert werden. Je größer die Distanz zwischen den Szintillatorplatten gewählt wird, desto kleiner wird der betrachtete Raumwinkel der koinzident messenden Detektoren. Hierbei ist es wichtig, die Trigger-Zeit den entsprechenden Abständen anzupassen. Diese wird aus dem Maximalabstand d zwischen den äußersten Detektoren über die Beziehung $T_{\text{Trigger}} = d/30 \frac{\text{cm}}{\text{ns}}$ ermittelt. Um die Detektoren in verschiedenen Abständen übereinander platzieren zu können, müsste noch eine geeignete Befestigungsmethode gefunden werden. Denkbar wäre auch ein Regal mit Zwischenteilen, welche auf verschiede-

nen Höhen eingesetzt werden können. Zur Messung könnten die Detektoren dann in diesen Fächern sicher abgelegt werden. Bei den Koinzidenzeinstellungen wird seitens der QuarkNet2-Betreiber die Zwei-Kanal-Koinzidenz empfohlen. Die Gate-Breite sollte ebenfalls im Bereich der Trigger-Zeit liegen.

3.2.2 Daten und Analyse

Leider musste viel Messzeit dem „Troubleshooting“ geopfert werden. Der hier vorgestellte Datensatz vom 21. und 22. April 2010 (Set I, Kanal 2) erfasst daher nur die Myonenrate über eine Dauer von 19 Stunden (siehe Abbildung 3.5). Die mittlere Myonenrate

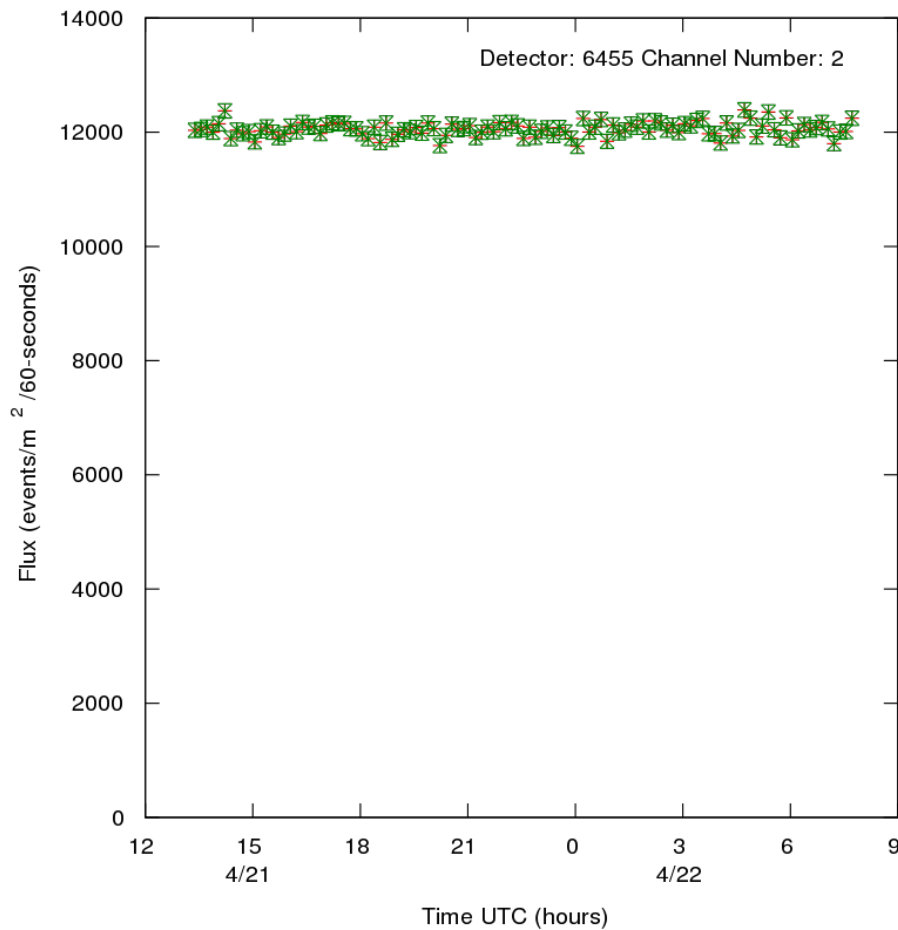


Abbildung 3.5: Myonenrate am 21./22. April 2010 (CReLab)

liegt bei $12076,4 \pm 11,7$ Events/m² min. Als Literaturwert wird eine Myonenrate von ~ 12000 Events/m² min [Heb01] angegeben. Da die hier verwendete Anordnung nur einen gewissen Raumwinkel betrachtet, muss dies in der Auswertung berücksichtigt werden (Genaueres in der Fehlerdiskussion).

3 Versuche mit dem Cosmic Ray Muon Detector

Bei den Messungen der Myonenrate traten eine Reihe von Fehlern auf, welche nun erläutert werden. Den QuarkNet2-Betreibern sind Probleme dieser Art bekannt. Daher empfehlen sie, die automatische Statuszeile bei der Datennahme einzuschalten. Sie wird über den DAQ-Board-Befehl **ST 2 5** aktiviert. Damit werden alle fünf Minuten eine Statuszeile und die aktuellen Zählerstände ausgegeben. Mit diesen Daten kann überprüft werden, ob der Detektor normal läuft.

Stufen

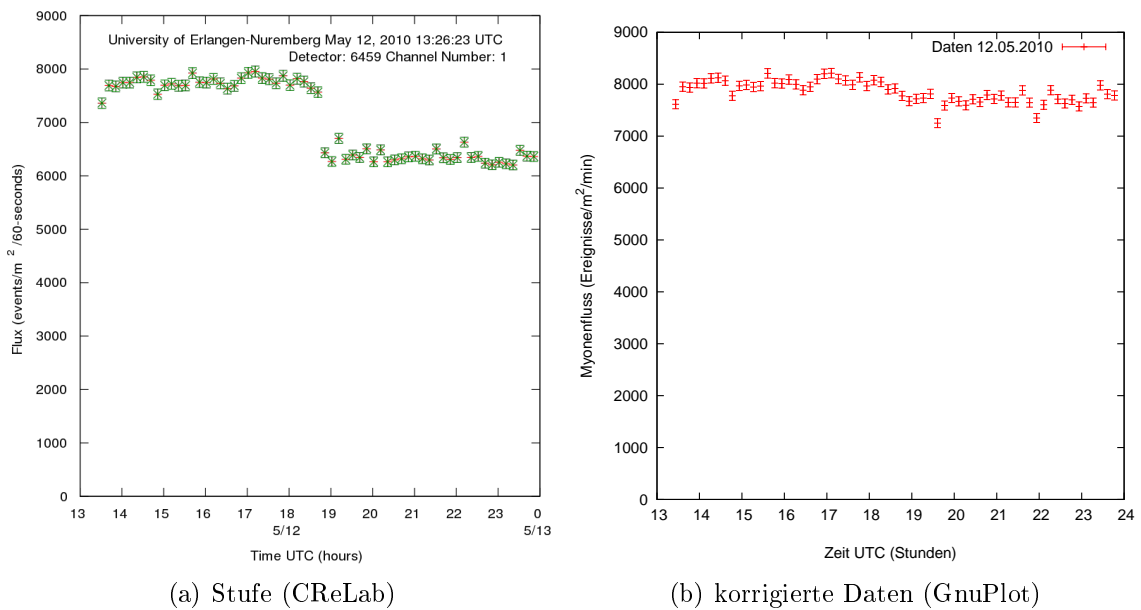


Abbildung 3.6: Daten einer fehlerhaften Myonenfluss-Messung (12. Mai 2010, Set II, Kanal 1)

Die am häufigsten auftretenden Probleme sind Störungen der Datennahme mit dem Terminalprogramm. Dabei werden meist Datenzeilen abgeschnitten und Informationen gehen verloren. Dieser Fehler tritt bei Verwendung des Windows Hyperterminal sehr oft auf. Da das Auftreten der Störung keine Systematik zeigt, scheint es eher zufällig verteilt zu sein. Um dies zu vermeiden wurde das Computerprogramm **getdata** (siehe Abschnitt 4.2.1 geschrieben, bei welchem Störungen dieser Art deutlich seltener vorkommen. Abbildung 3.6(a) zeigt den stufenförmigen Verlauf einer Myonenrate, falls bei der Datennahme Probleme auftreten. Oft besteht die Möglichkeit die Daten mit dem Java-Werkzeug **fill** (siehe Abschnitt 4.2.3) zu rekonstruieren. Das Ergebnis ist in Abbildung 3.6(b) zu sehen: Der Graph zeigt den gleichen Datensatz (Bin-Breite: 600 s) wie Abbildung 3.6(a), jedoch wurden hier die Daten vervollständigt. Dabei wurden 163705 der 1918858 Datenzeilen (Anteil 8,5%) rekonstruiert. Allerdings gingen bei der Rekonstruktion 11887 Datenzeilen (Anteil 0,6%) verloren.

Ausfall

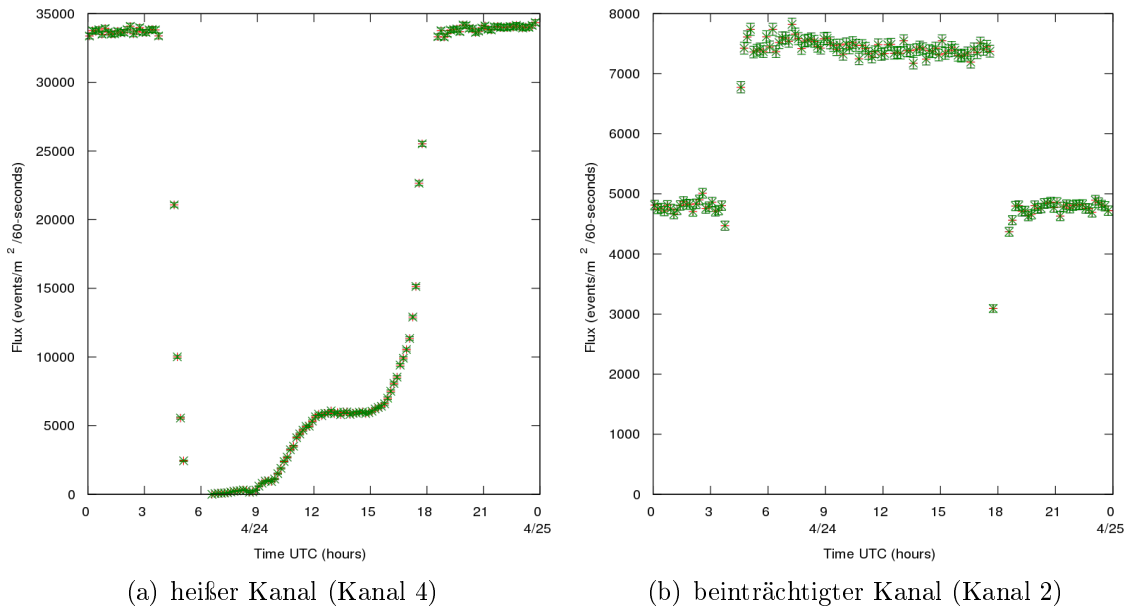


Abbildung 3.7: Gegenseitige Beeinflussung der Detektoren bei der Datenaufzeichnung (Daten vom 24. April 2010, Set II) (CReLab)

Steigt die gemessene Rate eines Detektors stark an, so kann dies zu einer Absenkung der Raten anderer Detektoren führen. Abbildung 3.7 zeigt ein entsprechendes Beispiel. In Abbildung 3.7(a) ist die sehr hohe Rate eines „heißen“ Detektors dargestellt. Als diese Rate kollabiert, steigt die Rate des Detektors in Abbildung 3.7(b) sprunghaft an. Seine Rate wurde bis dahin durch den heißen PMT unterdrückt. Dies liegt an der begrenzten Aufnahmekapazität des DAQ-Boards. Sollte sich die Charakteristik eines heißen Detektors nicht durch Trennung der Spannungsversorgung wieder normalisieren lassen, muss er auf Fehler (z.B. Lichtlöcher) untersucht und gegebenenfalls neu kalibriert werden. Manchmal können PMTs auch ohne ersichtlichen Grund einfach ausfallen. Dann sollten die Detektoren ebenfalls von der Spannungsversorgung getrennt werden. Nachdem wieder eine Spannung angelegt wird, zeigen die PMTs meist wieder ihr übliches Verhalten.

Unnatürliche Schwankungen

Von Zeit zu Zeit sind starke Schwankungen in den Myonenraten (ähnlich Abbildung 3.8) zu beobachten, die nicht auf Datenverlust zurückgeführt werden können. Der Verlauf dieser Schwankungen ist stetig, nicht stufenförmig wie bei einer Störung der Datennahme. Somit ist der Grund für das sonderbare Verhalten wohl bei den PMTs zu

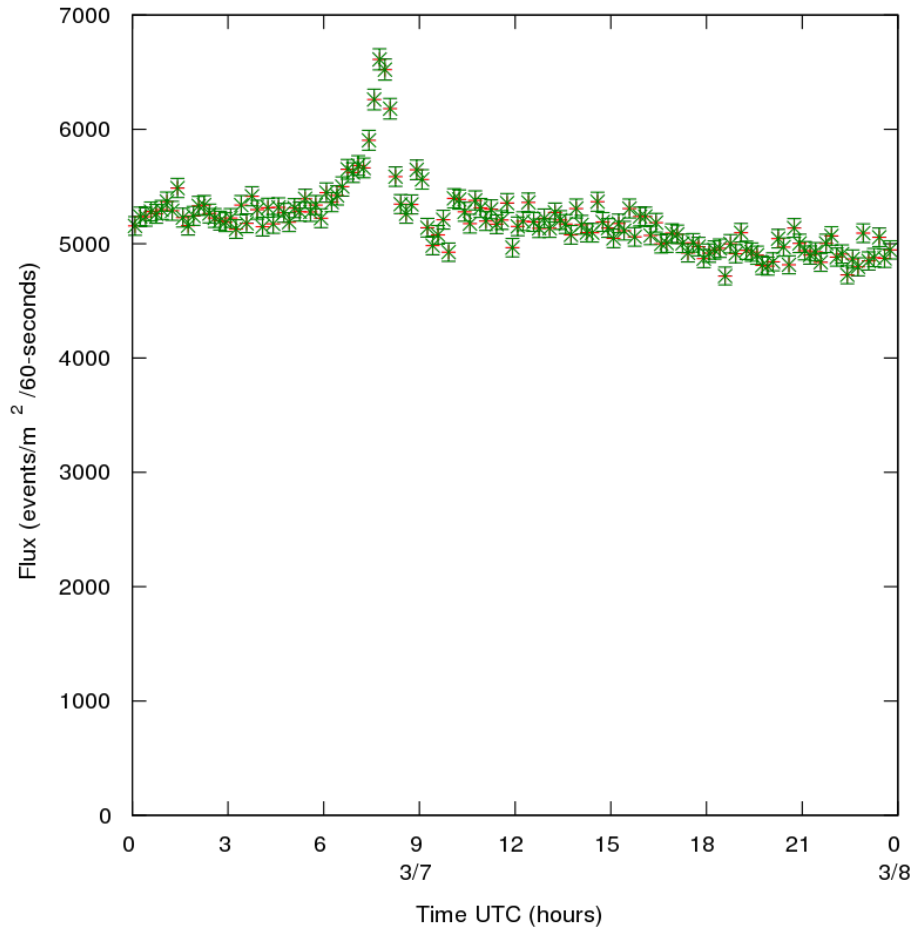


Abbildung 3.8: Unnatürliche Ratenveränderung in den Daten vom 7. März 2010 (Set II, Kanal 1) (CReLab)

suchen. Es wird vermutet, dass die PMTs ihre Charakteristiken bei Hitzeentwicklung (z.B. durch Sonneneinstrahlung) verändern und so die Ratenschwankungen zustande kommen. Auch könnten Heizkörper mit Nachtabstaltung und dadurch entstehende Temperaturdifferenzen zu einem solchen Verhalten führen. Deshalb sollten diese Fehlerquellen stets bei der Standortwahl des Detektors (Verdunklungsmöglichkeit, etc.) berücksichtigt werden.

3.2.3 Fehler

Die Intensitätsverteilung I der kosmischen Strahlung folgt der Winkelabhängigkeit $I \propto \cos^2 \theta$. Um den mit dem Detektorset betrachteten Zenitwinkelbereich zu erhalten, müssen die Abmessungen der Detektoren und die Anordnung bekannt sein. Die Szintillatorplatten haben eine Größe von $30 \text{ cm} \times 25 \text{ cm}$. Der Abstand der beiden äußeren Detektoren der hier verwendeten gestapelten Anordnung beträgt 12 cm . Daraus

3 Versuche mit dem Cosmic Ray Muon Detector

ergibt sich ein beobachtbarer Zenitwinkel von $\theta_{min} = \arctan \frac{25}{12} = 64,4^\circ$ für die kurze Kante und $\theta_{max} = \arctan \frac{\sqrt{25^2+30^2}}{12} = 72,9^\circ$ für die Diagonale. Damit kann nur ein Raumwinkel [Wikb] von

$$\Omega_{Detektor} = 4 \arcsin(\sin \theta_{min} \sin \theta_{max}) \text{ sr} = 1,32 \cdot \pi \text{ sr}$$

abgedeckt werden. Das entspricht einem Bruchteil von $\frac{1,32}{2} = 66,2\%$ des Himmels (Raumwinkel $\Omega = 2\pi \text{ sr}$).

Dennoch liegt der Anteil der theoretisch detektierbaren Myonen aufgrund der $\cos^2 \theta$ -Intensitätsverteilung höher. Es wird der mittlere Öffnungswinkel von $\bar{\theta} = \frac{\theta_{min} + \theta_{max}}{2} = 68,7^\circ$ zur Berechnung des Skalierungsfaktors s der Myonenrate über

$$\begin{aligned} s \cdot \int_{0^\circ}^{90^\circ} \cos^2 \theta \, d\theta &= \int_{0^\circ}^{68,7^\circ} \cos^2 \theta \, d\theta \\ s \cdot \frac{\pi}{4} &= \frac{1}{2} \left(\frac{68,7^\circ \cdot \pi}{180^\circ} + \cos(68,7^\circ) \sin(68,7^\circ) \right) \\ \Rightarrow s &= 0,979 \end{aligned}$$

verwendet. Obwohl also nur 66,2 % des Himmels überwacht werden können, können trotzdem theoretisch 97,8 % der Myonen detektiert werden (bei Ausrichtung der Detektoren parallel zum Erdboden). Wird der Literaturwert von 12000 Events/m² min mit dem Faktor s skaliert, ergibt sich eine erwartete Myonenrate von 11736 Events/m² min für den hier verwendeten Aufbau.

Zur genaueren Ratenbestimmung müsste zusätzlich auch die Verkleinerung der Detektorfläche bei schräg einfallenden Myonen berücksichtigt werden. Wahrscheinlich sind dazu Monte-Carlo-Simulationen analytischen Methoden vorzuziehen. Hier wird stets von einer konstanten Detektorfläche von 774,2 cm² für alle Richtungen ausgegangen. Trotz Koinzidenzmessung sind immer noch Untergrundereignisse zu verzeichnen, deren Rate ebenfalls abzuschätzen ist. Außerdem liegt die Detektionseffizienz der Szintillatoren nicht bei 100%.

3.3 Messung der Myonen-Lebensdauer

Die Lebensdauer von Myonen kann mit dem **CRMD** bestimmt werden. Dabei eignet sich der Versuch sehr gut, um den SchülerInnen die spezielle Relativitätstheorie anhand der Zeitdilatation experimentell vor Augen zu führen. Die Myonen der kosmischen Strahlung entstehen durch Zerfälle von Pionen, welche durch das Auftreffen von Primärteilchen auf die Erdatmosphäre ausgelöst werden.

$$\begin{aligned}\pi^+ &\longrightarrow \mu^+ + \nu_\mu \\ \pi^- &\longrightarrow \mu^- + \bar{\nu}_\mu.\end{aligned}$$

Die daraus resultierenden Myonen zerfallen wiederum nach dem Schema:

$$\begin{aligned}\mu^+ &\longrightarrow e^+ + \nu_e + \bar{\nu}_\mu \\ \mu^- &\longrightarrow e^- + \bar{\nu}_e + \nu_\mu.\end{aligned}\tag{3.1}$$

Dabei beträgt die mittlere Lebensdauer der Myonen nach [Par08]

$$\tau_\mu = 2,197019 \pm 0,000021 \text{ } \mu\text{s}.\tag{3.2}$$

Trotz der geringen mittleren Lebensdauer legen die Myonen beachtliche Strecke zurück. Dies liegt an ihren hohen Geschwindigkeiten (nahe Lichtgeschwindigkeit c). Wird ein Teilchen angenommen, das sich mit Lichtgeschwindigkeit c über eine Dauer von τ_μ bewegt, so legt es nach klassischer Rechnung ungefähr 660 m zurück:

$$v = \frac{s}{t} \Rightarrow s = c \cdot \tau_\mu \approx 660 \text{ m}.$$

Diese Rechnung ist hier aufgrund der speziellen Relativitätstheorie nicht anwendbar. Danach gilt für ein Teilchen mit Geschwindigkeit v das Prinzip der Zeitdilatation:

$$T_0 = T \cdot \sqrt{1 - \frac{v^2}{c^2}}.$$

T ist dabei die Zeit, die im Ruhesystem vergangen ist. Die in einem System, das sich mit Geschwindigkeit v bewegt, vergangene Zeit ist T_0 . Die daraus resultierende Ausdehnung der Zeit, macht es nun möglich, schnelle kosmische Myonen auch am Erdboden zu messen.

3.3.1 Aufbau und Methoden

Zur Messung der Myonenlebensdauer wird ein Detektorset benötigt. Die vier Detektoren werden vertikal übereinander angeordnet („stacked setup“), wobei die Szintillatorplatten parallel zum Boden ausgerichtet sind (siehe Abbildung 3.9). Aufgrund ihrer Bauart ist es möglich, die Detektoren aufeinander zu stapeln (siehe Abbildung 3.10).

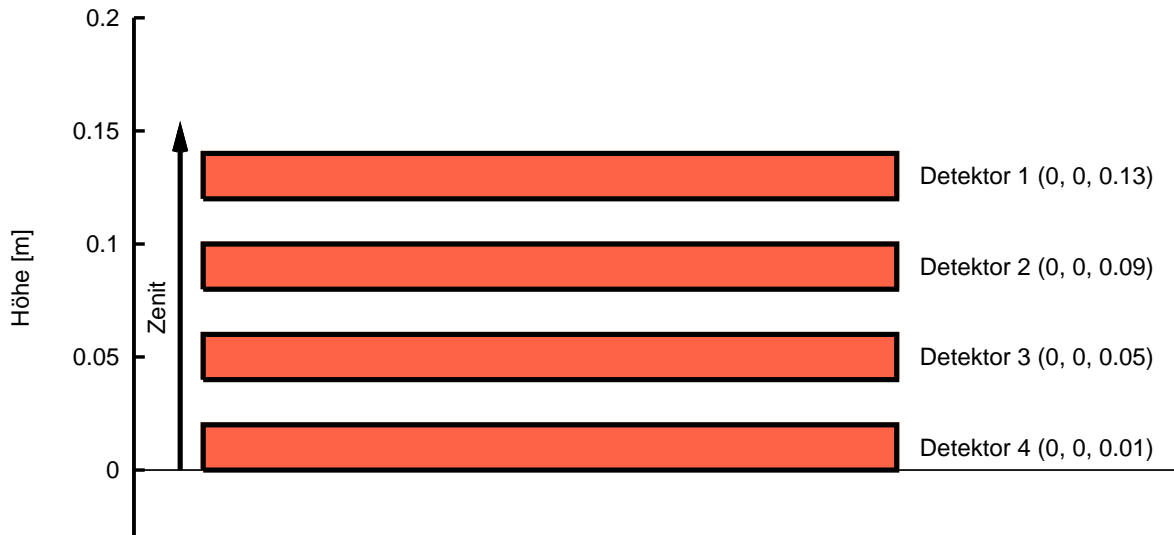


Abbildung 3.9: Anordnung der Detektoren zur Messung der Myonenlebensdauer (Gnu-Plot)



Abbildung 3.10: Gestapelte Detektoren zur Messung der Myonenlebensdauer

Die Detektorsets wurden an zwei Standorten betrieben (siehe Abbildung 3.11). Detektorset I wurde im Kontrollraum des Tandemlabors (49.580265°N, 11.027843°O) platziert. Set II wurde in Raum 108a (49.580821°N, 11.027797°O) betrieben. Beide Sets wurden wie in Abbildung 3.9 angeordnet.

Als Koinzidenzbedingungen werden von QuarkNet2 [Ber] zur Messung der Myonenlebensdauer folgende Parameter empfohlen:

3 Versuche mit dem Cosmic Ray Muon Detector

- Trigger-Zeit 48 ns
- Gate-Breite 9600 ns
- Zwei-Kanal-Koinzidenz

Das Ziel der Messung ist es nun, Myonen zu detektieren, die in einem der Detektoren eingefangen werden und anschließend zerfallen. Daher wird das Set in Zwei-Kanal-Koinzidenz betrieben und überprüft, welche Detektoren getroffen wurden. Wird im letzten getroffen Kanal kurz darauf ein weiteres Signal durch das Zerfallsprodukt e^\pm (siehe Gleichung (3.1)) gemessen, ohne dass zwischenzeitlich andere Detektoren koinzident getroffen wurden, ist dies ein Hinweis auf ein zerfallenes Myon. Aus den Zeitdifferenzen von Triggersignalen und Zerfallsignalen lässt sich eine Verteilung gewinnen:

$$I(t) = I_0 \cdot e^{-\frac{t}{\tau_\mu}}. \quad (3.3)$$

Mit den gewonnenen Daten und Gleichung (3.3) kann die mittlere Lebensdauer τ_μ der Myonen bestimmt werden.



Abbildung 3.11: Standorte der Messaufbauten (Google Maps)

Tabelle 3.1: Datensätze zur Messung der Myonenlebensdauer

Satz	Set	Zeitraum	Trigger-Zeit [ns]	Gate-Breite [ns]	Treffer
1	II	06. - 10. März 2010	80	16960	$8,5 \cdot 10^6$
2	II	06. - 11. Mai 2010	80	163840	$14,4 \cdot 10^6$
3	I	12. - 26. Mai 2010	80	163840	$26,4 \cdot 10^6$

3.3.2 Daten und Analyse

Insgesamt wurden drei Datensätze aufgenommen, deren Messzeitraum und Parameter Tabelle 3.1 zu entnehmen sind. Bei allen Datensätzen wurde Zwei-Kanal-Koinzidenz gewählt. Da anfangs von einer Zeitauflösung von 0,75 ns statt 1,25 ns ausgegangen wurde, weichen die Parameter Trigger-Zeit und Gate-Breite von den empfohlenen Werten (siehe Abschnitt 3.3.1) ab.

Aus den Rohdaten wurden die Zerfallszeiten mit Hilfe des „CR e-Lab“ [Fer] gewonnen. Dabei müssen drei Parameter eingestellt werden (siehe Abbildung 3.12). Das „Coincidence level“ gibt an, wieviele Detektoren getroffen werden müssen, um ein Event als möglichen Zerfall einzustufen. Je höher dieses gewählt wird, desto geringer wird der Untergrund (Anzahl zufälliger Events). Die „Gate width“ sagt dem Programm, wie lange es nach dem Trigger-Ereignis auf einen möglichen Zerfall warten soll. „Number of bins“ steht für die Anzahl gleich großer Teilintervalle, in welche das Zeitintervall [0 s; Gate width] zerlegt werden soll.

Nach abgeschlossenem Herausfiltern der Zerfallsereignisse durch die Online-Plattform können die Daten heruntergeladen werden. Die Datensätze können im Menü unter „Data -> Analyses“ ausgewählt werden. Anschließend kann das Datenverzeichnis über den Link „Show analysis directory“ aufgerufen werden. Die verarbeiteten Daten befinden sich in der Datei „frequencyOut“ und können nun lokal gespeichert werden. Die erste Spalte der Datei gibt die Zerfallsdauer in μs an, die zweite Spalte zeigt die Anzahl der Treffer des Intervalls an.

Die drei Datensätze wurden alle mit „Coincidence level“ zwei und einer „Gate width“ von 100 μs ausgewertet. Jedoch wurde die Auswertung für jeden Datensatz mit den „Binnings“ 40, 80 und 120 durchgeführt.

3 Versuche mit dem Cosmic Ray Muon Detector

[Understand the graph](#)

You're analyzing...	Chan1 events	Chan2 events	Chan3 events	Chan4 events	Raw Data	Remove from analysis
University of Erlangen- Nuremberg Mar 6, 2010 13:33:48 UTC	257948	321178	313124	277524	View Statistics Geometry	<input type="checkbox"/>
University of Erlangen- Nuremberg Mar 7, 2010 0:0:0 UTC	579590	739468	714119	650296	View Statistics Geometry	<input type="checkbox"/>
University of Erlangen- Nuremberg Mar 8, 2010 0:0:0 UTC	562220	733487	710516	631367	View Statistics Geometry	<input type="checkbox"/>
University of Erlangen- Nuremberg Mar 9, 2010 13:47:5 UTC	238321	332001	308852	253476	View Statistics Geometry	<input type="checkbox"/>
University of Erlangen- Nuremberg Mar 10, 2010 0:0:0 UTC	185435	253911	237865	193716	View Statistics Geometry	<input type="checkbox"/>
Total (5 files 8494414 events)	1823514	2380045	2284476	2006379	Compare files	<input type="button" value="Remove"/>

Analyze the same files in [flux](#) or [shower](#)

Click **Analyze** to use the default parameters. Control the analysis by expanding the options below.

▼ Analysis Controls

☐ Coincidence level:

☐ Gate width (seconds):

☐ Number of Bins:

► Plot Controls

► Fit Controls

▼ Execution Mode ☐

☒ Local (estimated time: 00:35:28)

☐ I2U2 Cluster (estimated time: 01:07:27)

☐ Grid (temporarily disabled)

☐ Automatic (estimated time: 00:49:37)

Abbildung 3.12: Einstellung der Parameter zur Myonen-Lebensdauer-Messung mit dem „CR e-Lab“

Datenfit 1

Die Daten, die mit dem „CRe-Lab“ bearbeitet wurden, wurden anhand der Funktion Gleichung (3.3) gefittet. Es ist jedoch zusätzlich noch ein Offset-Parameter c nötig, um den Untergrund zu erfassen:

$$I(t) = I_0 \cdot e^{-\frac{t}{\tau_\mu}} + c. \quad (3.4)$$

Der Fit mit dieser Funktion kann auch online im „CRe-Lab“ durchgeführt werden. Dazu muss bei der Auswertung im Unterpunkt „Fit Controls“ (siehe Abbildung 3.12)

Tabelle 3.2: Ergebnisse für die Myonenlebensdauer mit Fitmethode 1

Binning	40	80	120
Datensatz 1	$1,917 \pm 0,037 \mu\text{s}$	$1,954 \pm 0,053 \mu\text{s}$	$1,967 \pm 0,045 \mu\text{s}$
Datensatz 2	$2,193 \pm 0,040 \mu\text{s}$	$2,087 \pm 0,024 \mu\text{s}$	$2,088 \pm 0,022 \mu\text{s}$
Datensatz 3	$2,212 \pm 0,036 \mu\text{s}$	$2,188 \pm 0,021 \mu\text{s}$	$2,171 \pm 0,027 \mu\text{s}$

die entsprechende Option aktiviert werden. In Tabelle 3.2 sind die Myonenlebensdauern τ_μ der einzelnen Datensätze und Binnings für den Datenfit anhand von Gleichung (3.4) festgehalten. Diese Werte wurden mit GnuPlot, gewichtet nach ihren Fehlern, gemittelt, was zu einer mittleren Lebensdauer von $\tau_\mu = 2,112 \pm 0,031 \text{ } \mu\text{s}$ führte. Dieser Wert zeigt eine leichte Abweichung vom Literaturwert 3.2. In Abbildung 3.13 ist Datensatz 3 mit dem jeweils ermittelten Datenfit dargestellt.

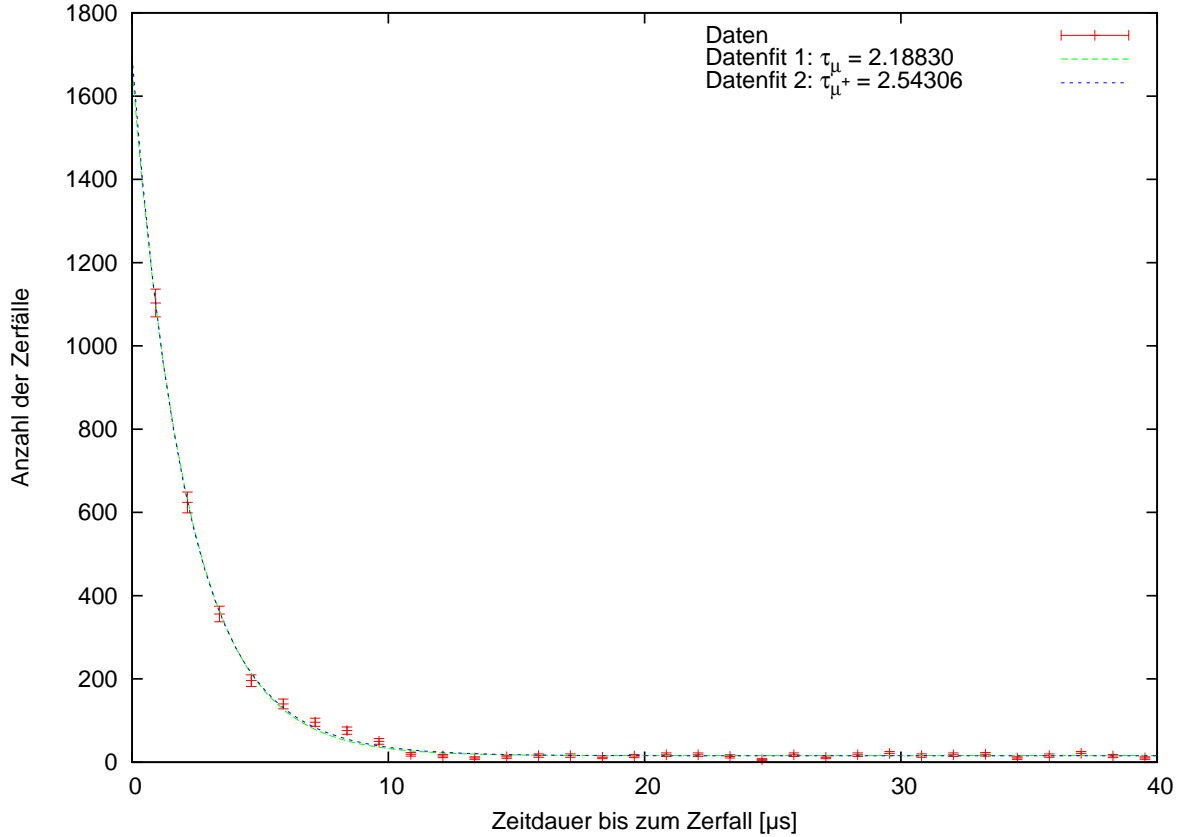


Abbildung 3.13: Datensatz 3 mit 80 bins (GnuPlot)

Datenfit 2

Es existiert eine weitere Reaktion, welche bei Verwendung von Szintillatormaterial mit großer Ordnungszahl ($Z \geq 6$) nicht mehr vernachlässigbar ist:



Es handelt sich hierbei um die „Myonen-Version“ des Elektronen-Einfangs. Bei niedrigerenergetischen (also vorher abgebremsten) μ^- ist die Einfangwahrscheinlichkeit $\propto Z^4$.

3 Versuche mit dem Cosmic Ray Muon Detector

In kosmischen Schauern werden freie positive und negative Myonen im Verhältnis

$$\frac{I_{0,\mu^+}}{I_{0,\mu^-}} = \frac{0,56}{0,44} \approx 1,27$$

erzeugt. Ihre mittleren Zerfallszeiten $\tau_{\mu^-} = \tau_{\mu^+} = \tau$ sind im freien Zustand gleich. In Materie jedoch ändert sich die Zerfallslänge des negativen Myons aufgrund des Myoneneinfangs (siehe Gleichung (3.5)). Ihr theoretischer Wert liegt bei $\tau_{\mu^-} = 1,7 \mu\text{s}$ [Ros52]. Dies muss beim Fitten der Daten zusätzlich berücksichtigt werden [SL07]:

$$I(t) = I_0 \cdot \left(0,44 \cdot e^{-\frac{t}{\tau_{\mu^-}}} + 0,56 \cdot e^{-\frac{t}{\tau_{\mu^+}}} \right) + c. \quad (3.6)$$

Die in Gleichung (3.6) angebrachte Korrektur führte bei den drei Datensätzen und jeweils fest auf $\tau_{\mu^-} = 1,7 \mu\text{s}$ gesetzter mittlerer Lebensdauer des μ^- zu den in Tabelle 3.3 aufgeführten Werten für die Myonenlebensdauer τ_{μ^+} . Diese Ergebnisse wurden ebenfalls mit GnuPlot gemittelt (unter Gewichtung der Fehler) und ergaben einen Wert von $\tau_{\mu^+} = 2,421 \pm 0,051 \mu\text{s}$ für die mittlere Myonenlebensdauer. Abbildung 3.14 zeigt Datensatz 1 mit 120 bins und die mit beiden Methoden berechneten Datenfits.

Tabelle 3.3: Ergebnisse für die Myonenlebensdauer mit Fitmethode 2

Binning	40	80	120
Datensatz 1	$2,064 \pm 0,060 \mu\text{s}$	$2,123 \pm 0,091 \mu\text{s}$	$2,151 \pm 0,078 \mu\text{s}$
Datensatz 2	$2,505 \pm 0,062 \mu\text{s}$	$2,383 \pm 0,037 \mu\text{s}$	$2,396 \pm 0,035 \mu\text{s}$
Datensatz 3	$2,544 \pm 0,053 \mu\text{s}$	$2,543 \pm 0,033 \mu\text{s}$	$2,529 \pm 0,045 \mu\text{s}$

Ruhmassenbestimmung des Myons

Aus der erhaltenen mittleren Lebensdauer des Myons kann auch seine Ruhemasse bestimmt werden. Sie ergibt sich mit der Fermi-Kopplungskonstante G_F aus der Beziehung

$$(m_\mu c^2)^5 = \frac{192\pi^3 \hbar^7 c^6}{G_F^2 \tau_\mu} \text{ mit } \frac{G_F}{(\hbar c)^3} = 1,16637 \cdot 10^{-5} \text{ GeV}^{-2}.$$

Wird hier der Wert $\tau_\mu = 2,151 \pm 0,078 \mu\text{s}$ (Datenfit 2, Datensatz 1, Binning 120) eingesetzt, so ergibt sich:

$$m_\mu c^2 = 106,103 \pm 0,765 \text{ MeV}.$$

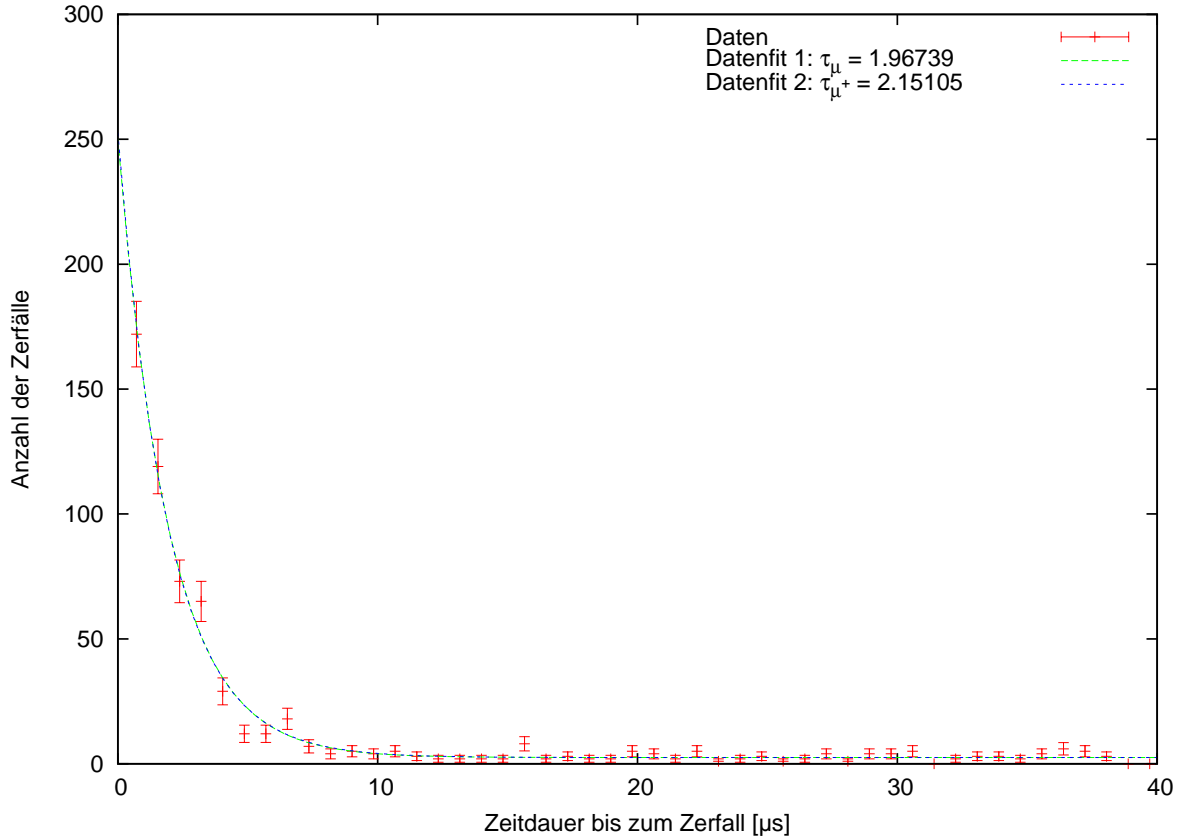


Abbildung 3.14: Datensatz 1 mit 120 bins (GnuPlot)

Der Literaturwert liegt bei $m_{\mu}c^2 = 105,658367 \pm 0,000004$ MeV [Par08]. Der gemessene Wert ist also im Rahmen der Messunsicherheit mit dem Literaturwert verträglich.

3.3.3 Fehler

Die Hauptfehlerquelle bei der Messung der Myonenlebensdauer steckt in den zufälligen Ereignissen, welche fälschlicherweise als Zerfall gewertet werden. Dieser Untergrund ist umso größer, desto höher die Gate-Breite auf dem DAQ-Board eingestellt wird. Bei den Datensätzen 2 und 3 wich der Parameter mit 163840 ns stark vom empfohlenen Wert von ≈ 10000 ns ab. Deshalb liegen die Werte beider Sätze mit der Fit-Methode 2 ($\in [2,383 \mu\text{s}, 2,544 \mu\text{s}]$) weit oberhalb des Literaturwerts ($2,197019 \mu\text{s}$). Da Fit-Methode 1 für „gute Daten“ stets zu kleine τ_{μ} -Werte liefert, werden „zufällig“ akzeptable Werte für die Datensätze 2 und 3 berechnet.

Eine weitere Ursache für hohen Untergrund ist eine schlechte Kalibration der Detektoren. Werden diese mit zu hoher Spannung oder zu niedriger Schwelle betrieben, ist das Untergrundrauschen stark ausgeprägt.

3 Versuche mit dem Cosmic Ray Muon Detector

Weitere Fehler entstehen durch das „Binning“ der Daten. Hier ist keine eindeutige Aussage zu treffen, für welche Größe der Fit die besten Werte liefert. Tendenziell sollte eher ein größeres „Binning“ (≥ 100) gewählt werden. Binnings, die die Länge der Teilintervalle kleiner als die Zeitauflösung (1,25 ns) werden lassen, sind natürlich nicht sinnvoll.

Aufgrund dieser Überlegungen wird der Wert (Fitmethode 2)

$$\tau_\mu = 2,151 \pm 0,078 \text{ } \mu\text{s} \quad (\pm 3,61\%)$$

aus Datensatz 1 bei einem Binning von 120 als Endergebnis der Messung festgestellt. Der Fehler bewegt sich mit 3,61% im akzeptablen Bereich für die Versuchsanordnung.

3.4 Schauereperimente

Ein Ziel des Schulprojekts ist der Nachweis von Teilchenschauern, die beim Aufprall hochenergetischer Primärteilchen auf die Erdatmosphäre entstehen. Deshalb wurden hier bereits vorab Tests mit zwei Detektorsets durchgeführt.

3.4.1 Aufbau und Methoden

Zur Detektion von Teilchenschauern standen zwei CRMD-Sets bereit. Diese wurden an verschiedenen Standorten in ausgebreiteten Geometrien platziert. So ist es möglich, durch Wahl geeigneter Koinzidenzbedingungen Schauer zu messen und ihre Richtung zu rekonstruieren. Die Koinzidenzbedingungen bestehen aus drei Parametern, welche über Befehle an das DAQ-Board übermittelt werden: #-Kanal-Koinzidenz (Anzahl getroffener Kanäle), Trigger-Zeit (Zeitabstand, innerhalb dessen die Kanäle getroffen werden müssen) und Gate-Breite (Zeitdauer der Aufzeichnung nach dem Trigger). Von den QuarkNet2-Betreibern wird für Shower-Arrays 2-Kanal-Koinzidenz empfohlen [Ber], um die Datenmenge überschaubar zu halten. Außerdem sollte die Trigger-Zeit 1200 ns und die Gate-Breite 3600 ns betragen.

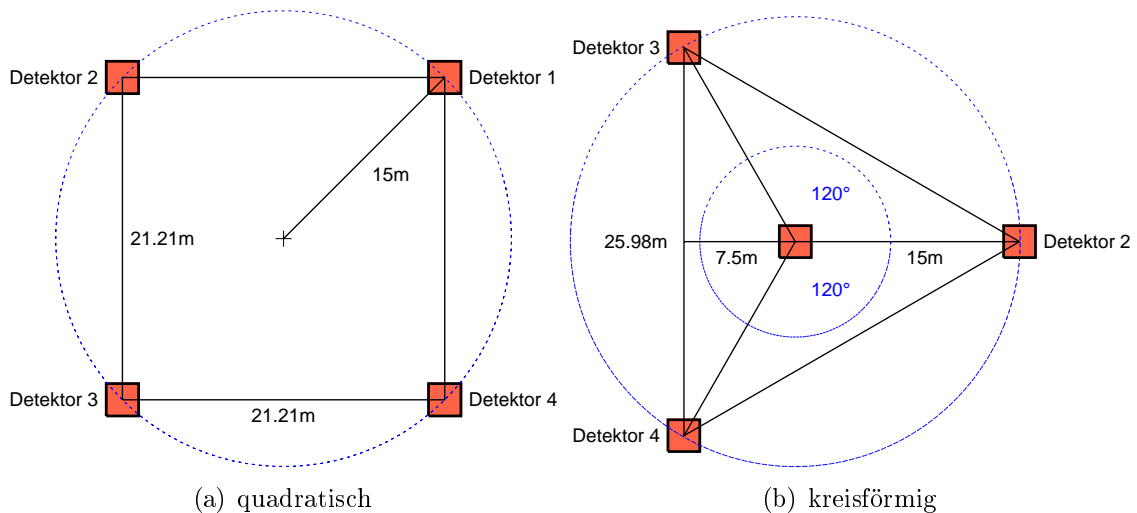


Abbildung 3.15: Möglichkeiten der Detektoranordnung (GeoGebra)

Die Länge der Spannungs- und Signalkabel beträgt ca. 15 m, das GPS-Kabel ist ca. 30 m lang. Das bedeutet bei einer quadratischen Anordnung eine maximale Seitenlänge von ca. 21 m (siehe Abbildung 3.15(a)). Als Anordnung wird vom Sky-View-Experiment [Heb01] empfohlen, einen Detektor im Mittelpunkt eines Kreises zu platzieren, wobei die restlichen drei in gleichen Abständen zueinander auf der Kreislinie

angeordnet werden (siehe Abbildung 3.15(b)). Das entspricht einem maximalen rechteckigen Platzbedarf von ca. $26\text{ m} \times 23\text{ m}$. Die Geometrie innerhalb eines Detektorsets musste stets den räumlichen Gegebenheiten angepasst werden. Die vier Detektoren wurden annähernd rechteckig aufgestellt. Im Schulexperiment dürften die räumlichen Strukturen ebenfalls ein Problem bei der Wahl der Detektorgeometrie darstellen. Da der große Platzbedarf wahrscheinlich nur auf Dächern oder Außenflächen realisierbar sein dürfte, müsste das Experiment wetterfest sein, um die maximalen Geometrien im Schulbetrieb auszureizen.

Aufbau 1: Physikalisches Institut - Tandemlabor

Der erste Aufbau mit beiden Detektorsets befand sich im Raum 108a des Physikalischen Instituts (Erwin-Rommel-Straße 1) und im Kontrollraum des Tandembeschleunigers (siehe Abbildung 3.16). Die beiden Sets hatten einen Abstand von ca. 60 m zueinander (ermittelt mit dem „BayernViewer“ der Bayerischen Vermessungsverwaltung). Die Geo-



Abbildung 3.16: Aufbau 1 (Google Maps)

metrien der beiden Detektorsets wurden den räumlichen Gegebenheiten angepasst. Die Koordinaten sind als (x,y,z) in Metern zu messen, wobei die x -Achse Richtung Osten, die y -Achse Richtung Norden und die z -Achse Richtung Zenit zeigt (Kompass zur Ausrichtung benötigt). Breiten- und Längengrade der Detektoren wurden mit Google Maps ermittelt, da die GPS-Werte der DAQ-Boards starken Schwankungen unterlagen. Set I war im Tandemlabor (49.580265°N , 11.027843°O) angeordnet wie in Abbildung 3.17(a)

3 Versuche mit dem Cosmic Ray Muon Detector

dargestellt. Set II befand sich im Raum 108a (49.580821°N , 11.027797°O) und wurde aufgebaut wie in Abbildung 3.17(b) dargestellt. Set I bestand zu diesem Zeitpunkt lediglich aus drei Detektoren. Die Szintillatorplatten waren jeweils parallel zum Erdboden ausgerichtet.

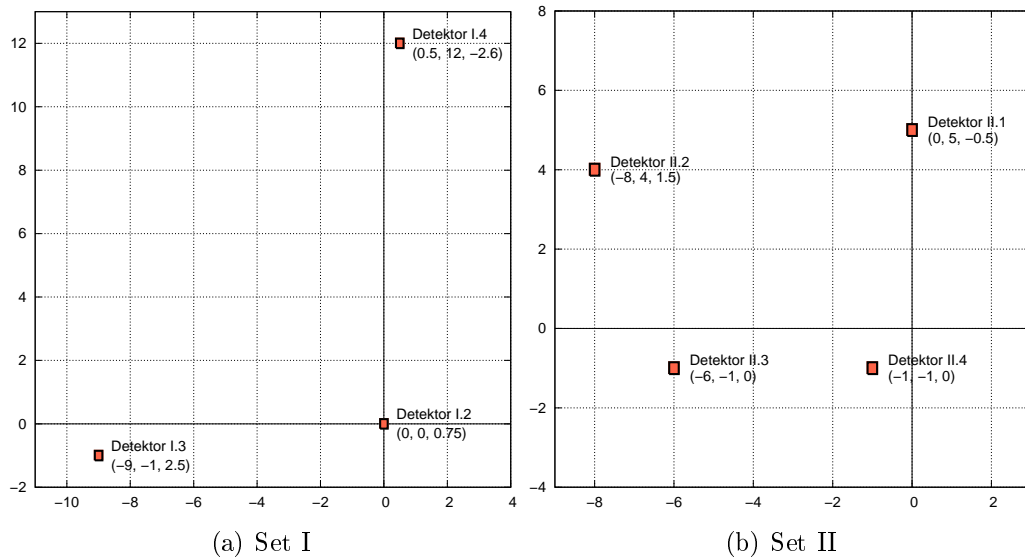


Abbildung 3.17: Detektoranordnungen Aufbau 1 (GnuPlot)

Aufbau 2: Hörsaalgebäude (Vorbereitung HD) - Tandemlabor

Für den zweiten Aufbau der Detektorsets wurden als Standorte der Vorbereitungsraum des Hörsaal D (Hörsaalgebäude) und der Kontrollraum des Tandembeschleunigers gewählt (siehe Abbildung 3.18). Bei diesem Aufbau hatten die beiden Sets einen Abstand von ca. 218 m zueinander (BayernViewer). Dabei wurde Set I im Tandemlabor (49.580265°N , 11.027843°O) angeordnet wie in Abbildung 3.19(a) dargestellt. Set II wurde im Vorbereitungsraum des HD (49.579583°N , 11.030657°O) platziert, wie in Abbildung 3.19(b) gezeigt.

3.4.2 Daten und Analyse

Die Auswertung geschah mit einer Reihe selbst geschriebener Tools. Zur Auswertung kann aber auch das „Cosmic Ray e-Lab“ verwendet werden. Mit dessen Hilfe ist es möglich, Koinzidenzen in einem und zwischen mehreren Detektorsets zu identifizieren. Dabei muss zuerst definiert werden, was ein Ereignis ist. Dies geschieht anhand der

3 Versuche mit dem Cosmic Ray Muon Detector



Abbildung 3.18: Aufbau 2 (Google Maps)

vier Parameter: Event Gate, Detector Coincidence, Channel Coincidence und Event Coincidence.

Detector Coincidence	Anzahl der Detektorsets, die getroffen werden müssen.
Channel Coincidence	Anzahl der Detektoren aller Detektorsets, die getroffen werden müssen.
Event Coincidence	Anzahl der nötigen Treffer aller Detektoren.
Event Gate	Zeitspanne in ns , während der all diese Bedingungen erfüllt sein müssen, damit die Detektortreffer als Ereignis eingestuft werden.

Um Ereignisse mit sämtlichen Einfallswinkeln zu erfassen, wird das Event Gate aus dem Grenzfall des streifenden Einfalls ($\phi = 90^\circ$ siehe Abbildung 3.20) abgeschätzt zu:

$$\text{Event-Gate} = \frac{\text{Abstand Detektorsets}}{\text{Lichtgeschwindigkeit}} \approx \text{Abstand Detektorsets} \cdot 3 \text{ ns/m.} \quad (3.7)$$

Dabei wird von einer vernachlässigbaren Schauerdicke ausgegangen.

Koinzidenzen zwischen beiden Detektorsets

Zur Messung ausgedehnter Luftschauer werden Koinzidenzen zwischen den beiden Detektorsets betrachtet. Dazu wurden Aufbau 1 und 2 verwendet. Für die Berechnung von Schauerflächen am Erdboden sind Simulationen (z.B. mit CORSIKA, siehe auch

3 Versuche mit dem Cosmic Ray Muon Detector

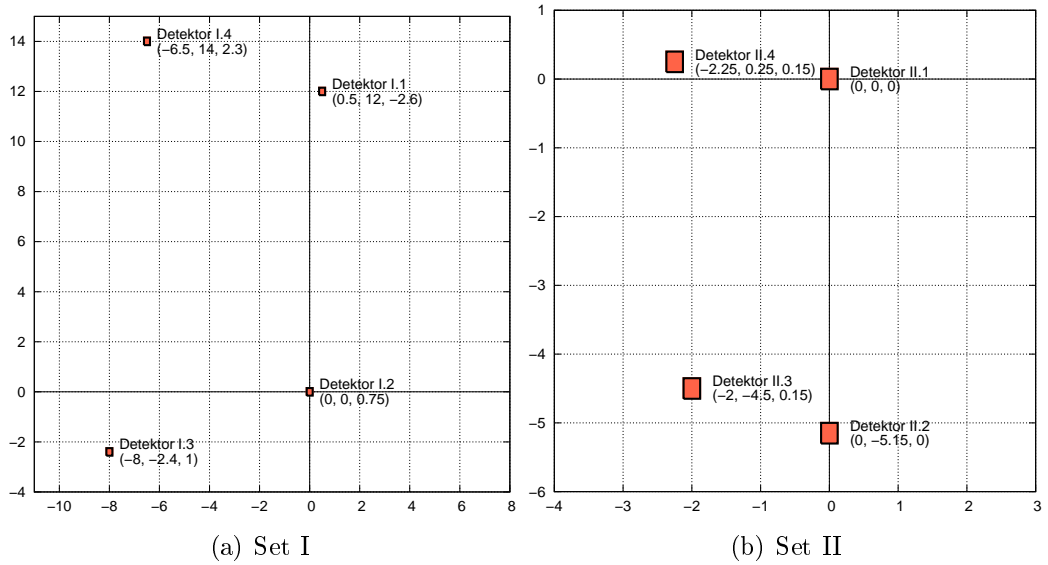


Abbildung 3.19: Detektoranordnungen Aufbau 2 (GnuPlot)

Kapitel 5) nötig, da sie von mehreren Parametern (Interaktionshöhe, Primärteilchentyp, etc.) abhängen. Jedoch ist die Schauerfläche (feste Interaktionshöhe, bestimmtes Primärteilchen, fester Einfallswinkel) proportional zur Energie des Teilchens. Die Häufigkeit der Primärteilchen hoher Energie fällt jedoch mit etwa $E^{-2.7}$ bis $E^{-3.0}$, je nach Energiebereich, ab.

Der Detektorsetabstand von ca. 60 m liefert mit Gleichung (3.7) ein Event-Gate von ca. 180 ns für Aufbau 1. Als Trigger-Zeit wurden 600 ns eingestellt, die Gate-Breite betrug 6000 ns. Die Detektoren wurden in 2-Kanal-Koinzidenz betrieben. Dabei wurden im Zeitraum vom 2. April bis 22. April 2010 zwei Koinzidenzen zwischen beiden Sets festgestellt. Das erste Schauerereignis traf am 8. April 2010 um 3:52:48 UTC insgesamt vier Detektoren der beiden CRMD-Sets. Neun PMTs haben ebenfalls am 8. April 2010

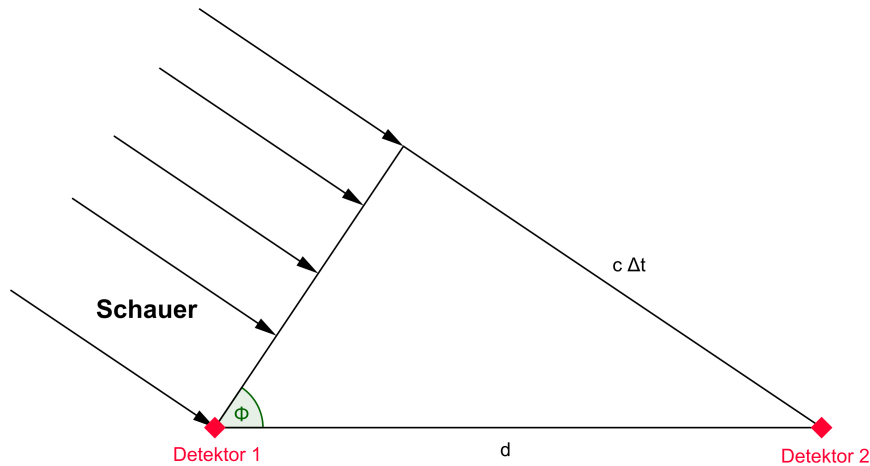


Abbildung 3.20: Zusammenhang zwischen Schauerwinkel ϕ und Δt (GeoGebra)

um 13:46:02 UTC koinzidente Ereignisse registriert.

Für Aufbau 2 ergibt sich mit Gleichung (3.7) und Detektorsetabstand ca. 218 m ein Event-Gate von ca. 660 ns. Es wurden für beide Sets die gleichen Parameter eingestellt, welche von QuarkNet2 für ein Detektor-Array empfohlen werden. Im Zeitraum vom 2. Juni bis 30. Juni 2010 wurden keine Koinzidenzen zwischen den Sets beobachtet, was aufgrund der kurzen Beobachtungsdauer zu erwarten war.

Rekonstruktion der Schauerrichtung

Die Rekonstruktion der Schauerrichtung ist erst möglich, wenn bei einem Event mindestens drei Detektoren getroffen wurden (drei Vektoren legen eine Ebene eindeutig fest). Die erforderlichen Daten sind dabei die Koordinaten der Detektoren (x = Ost-West, y = Nord-Süd, z = Zenit-Nadir) und die Zeiten der Treffer. Die Idee ist, die Schauerrichtung als Normalenvektor der Schauerebene aufzufassen. Eine Ebene, die durch den Ursprung verläuft, wird von zwei Vektoren aufgespannt. Diese müssen aus den Daten für jedes Event gewonnen werden. Dazu werden die Abstände der getroffenen Detektoren und die Zeitdifferenzen der Treffer errechnet. Die Neigungswinkel der beiden Ebenenvektoren gegenüber dem Erdboden werden über die Beziehung

$$\sin \phi_i = \frac{c \cdot \Delta t_i}{d_i} \quad (3.8)$$

ermittelt (siehe Abbildung 3.20), wobei c die Lichtgeschwindigkeit, d den Abstand der Detektoren und Δt die Zeitdifferenz zwischen den Treffern darstellt. Anschließend werden die Differenzenvektoren der Orte $\vec{\Delta d}_i$ der getroffenen Detektoren ermittelt und um die Winkel ϕ_i aus Gleichung (3.8) gedreht. Die normierten Rotationsachsen \vec{r}_{0i} bekommt man aus dem Kreuzprodukt der Ortsdifferenzen $\vec{\Delta d}_i$ und der z -Achse:

$$\vec{r}_{0i} = \frac{\vec{r}_i}{\|\vec{r}_i\|} = \frac{\vec{\Delta d}_i \times (0, 0, 1)^T}{\|\vec{\Delta d}_i \times (0, 0, 1)^T\|} = \frac{(\Delta d_{yi}, -\Delta d_{xi}, 0)^T}{\sqrt{(\Delta d_{yi})^2 + (\Delta d_{xi})^2}} = (r_{xi}, r_{yi}, 0)^T.$$

Zur Rotation der Differenzenvektoren wird die allgemeine Form der Rotationsmatrix [Wika] verwendet:

$$M_{rot}(\vec{r}_{0i}, \phi_i) = \begin{pmatrix} \cos \phi_i + r_{xi}^2(1 - \cos \phi_i) & r_{xi}r_{yi}(1 - \cos \phi_i) - r_{zi} \sin \phi_i & r_{xi}r_{zi}(1 - \cos \phi_i) + r_{yi} \sin \phi_i \\ r_{yi}r_{xi}(1 - \cos \phi_i) + r_{zi} \sin \phi_i & \cos \phi_i + r_{yi}^2(1 - \cos \phi_i) & r_{yi}r_{zi}(1 - \cos \phi_i) - r_{xi} \sin \phi_i \\ r_{zi}r_{xi}(1 - \cos \phi_i) - r_{yi} \sin \phi_i & r_{zi}r_{yi}(1 - \cos \phi_i) + r_{xi} \sin \phi_i & \cos \phi_i + r_{zi}^2(1 - \cos \phi_i) \end{pmatrix}.$$

3 Versuche mit dem Cosmic Ray Muon Detector

Damit ergibt sich:

$$\begin{aligned}\vec{v}_1 &= M_{rot}(\vec{r}_{01}, \phi_1) \cdot \vec{\Delta d}_1 \\ \vec{v}_2 &= M_{rot}(\vec{r}_{02}, \phi_2) \cdot \vec{\Delta d}_2.\end{aligned}$$

Der Normalenvektor \vec{n} der Ebene und somit die Richtung des Schauers errechnet sich aus dem normierten Kreuzprodukt der beiden Vektoren \vec{v}_1 und \vec{v}_2 , die die Ebene aufspannen, zu:

$$\vec{n} = \frac{\vec{v}_1 \times \vec{v}_2}{\|\vec{v}_1 \times \vec{v}_2\|}.$$

Die Richtungskordinaten im Horizontsystem, also den Azimutwinkel a und die Höhe h , werden aus dem Normalenvektor berechnet. Der astronomische Azimut gibt die Himmelsrichtung wieder (Süden $\hat{=}$ 0° , Westen $\hat{=}$ 90° , Norden $\hat{=}$ 180° , Osten $\hat{=}$ 270°), während die Höhe für den Winkel zwischen Horizont und beobachtetem Objekt steht ($h \in [0^\circ, 90^\circ]$).

$$\begin{aligned}\sin h &= \frac{n_z}{1} \\ \tan(270^\circ - a) &= \frac{n_y}{n_x}\end{aligned}$$

Dabei muss wegen der tan-Funktion noch eine Fallunterscheidung durchgeführt werden, um den vollen Wertebereich von -90° bis 270° abzudecken. Der erhaltene Wert wird anschließend von 270° abgezogen, damit Drehsinn und Nullpunkt ($0^\circ \hat{=}$ Süden) korrekt sind.

Diese Daten können verwendet werden, um die Ost-West-Abhängigkeit und die Winkelabhängigkeit der Myonen zu überprüfen.

Um Korrelationen von Schauerrichtungen mit kosmischen Objekten zu betrachten, müssen die Daten vom Horizontsystem in das ruhende und anschließend in das rotierende Äquatorsystem transferiert werden. Die Umrechnung in das ruhende Äquatorsystem erfordert den Azimut a , die Höhe h , aus welcher sich die Zenitdistanz $z = 90^\circ - h$ ergibt, und die geografische Breite φ (Nord-Süd). Um den Stundenwinkel t und die Deklination δ zu erhalten, sind drei Gleichungen [Grec] zu lösen:

$$\begin{aligned}\cos \delta \cos t &= \sin z \cos a \sin \varphi + \cos z \cos \varphi \\ \cos \delta \sin t &= \sin z \sin a \\ \sin \delta &= -\sin z \cos a \cos \varphi + \cos z \sin \varphi.\end{aligned}$$

Zur Transformation vom ruhenden in das rotierende Äquatorsystem wird die lokale

3 Versuche mit dem Cosmic Ray Muon Detector

Sternzeit θ benötigt [Greb]. Diese wird aus dem Julianischen Datum [Grea], der UT (in Dezimalstunden) und der geografischen Länge λ (Ost-West) berechnet:

$$\begin{aligned} JD_0 &= [365,25(Y + 4716)] + [30,6001(M + 1)] + D - 1524,5 \\ \Theta &= 6^h,664520 + 0^h,0657098244(JD_0 - 2451544,5) + 1^h,0027379093 \cdot UT \\ \theta &= \Theta + \lambda \frac{1^h}{15^\circ}. \end{aligned}$$

Y, M und D sind Jahr, Monat und Tag des benötigten gregorianischen Datums. Θ ist die mittlere Sternzeit in Greenwich (Nullmeridian). $[]$ steht für die Gaußsche Abrundungsfunktion. Die Deklination δ ändert sich nicht, während der Stundenwinkel t in die Rektaszension α übergeht:

$$\alpha = \theta - t.$$

Diese Umrechnungen werden von einem Java-Programm (**anglet**) durchgeführt. Die Ausgabedatei enthält Azimut, Höhe, Rektaszension und Deklination der Schauer gespeichert. Dabei rekonstruiert das Programm die Richtung von Schauern, die drei oder vier unterschiedliche Detektoren getroffen haben (Channel und Event Coincidence). Die Ergebnisse der Messungen vom 3. - 29. Juni 2010 mit Detektorset II sind in Abbil-

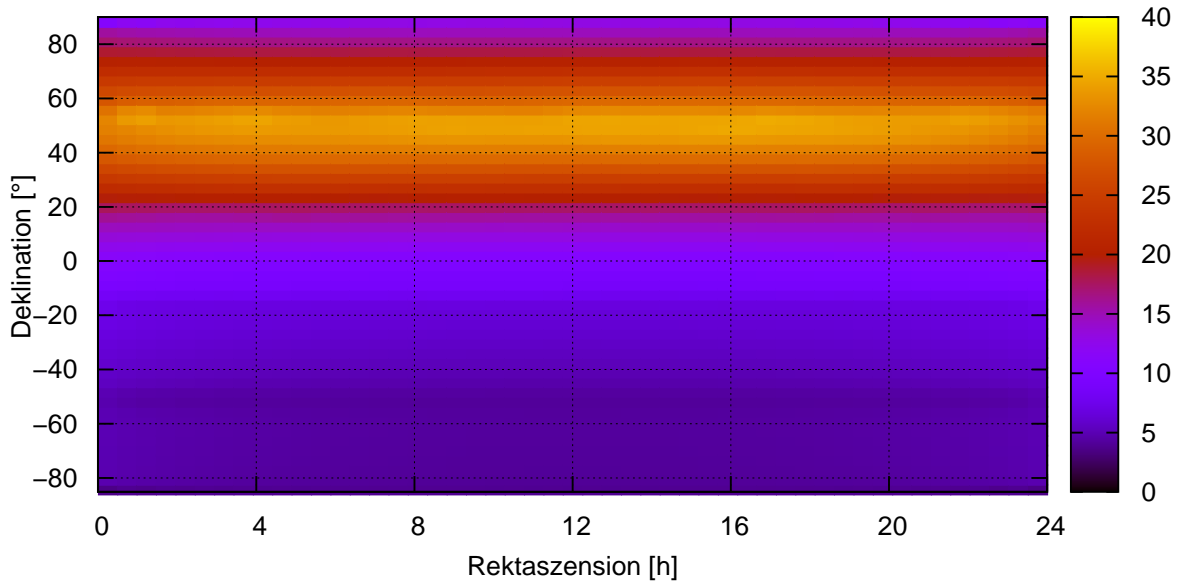


Abbildung 3.21: Richtungsverteilung der Luftschauer gemessen mit Detektorset II (GnuPlot)

Abbildung 3.21 zu sehen. Hierfür wurden 29974 Schauerereignisse ausgewertet, welche mit einer durchschnittlichen Rate von $\approx 9 \frac{\text{Schauer}}{\text{min}}$. Die Detector Coincidence war eins, da nur Set II betrachtet wurde. Dabei wurde ein Binning von $\Delta\alpha = 0,5^h$ für die Rektaszension und $\Delta\delta = 4^\circ$ für die Deklination zwischen -45° und 90° gewählt. Während ein Maximum der Schauerhäufigkeit für eine bestimmte Rektaszension in den Daten nicht auszumachen ist, ist bei einer Deklination von $\delta \approx 49^\circ$ eine deutliche Erhöhung

zu sehen. Dieser Wert entspricht dem Zenitwinkel $z = 0^\circ$ für unseren Messstandort. Da die Verteilung I der kosmischen Myonen sich wie $I \propto \cos^2 z$ verhält, war eine solche Häufigkeitsverteilung zu erwarten. Wird diese Abhängigkeit berücksichtigt, so ist keine Vorzugsrichtung zu erkennen.

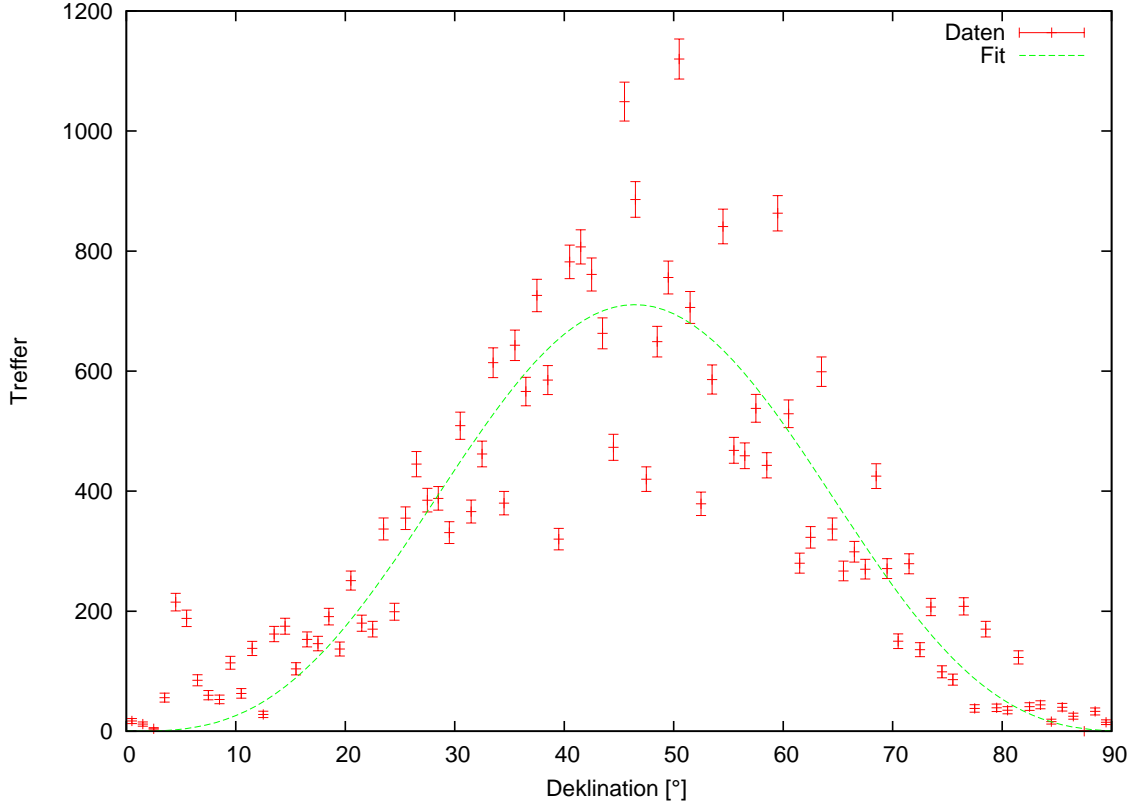


Abbildung 3.22: Winkelabhängigkeit der Schauerhäufigkeit (GnuPlot)

Deutlicher wird die Winkelabhängigkeit, wenn die Treffer über sämtliche Rektaszensionen aufsummiert (Binning $\Delta\alpha = 24^h$) und über der Deklination (Binning $\Delta\delta = 1^\circ$) aufgetragen werden (siehe Abbildung 3.22), was einer Projektion auf die Deklinations-Achse entspricht. Da mit zunehmendem Zenitwinkel die Detektorfläche kleiner wird muss dies ebenfalls durch $I \sim |\cos z|$ berücksichtigt werden. Der Fit in Abbildung 3.22 lautet daher:

$$I(\delta) = 710,761 \cdot \left| \cos^3 \left(\frac{(\delta - 46,44^\circ) \cdot 90^\circ}{46,44^\circ} \right) \right|.$$

Die höchste Trefferrate liegt somit bei einer Deklination von $\delta \approx 46,44^\circ \pm 0,67^\circ$.

3.4.3 Fehler

Die Richtungsrekonstruktion der Schauer leidet unter mehreren Messungenauigkeiten. Da die GPS-Daten der DAQ-Boards aufgrund der ständig wechselnden Anzahl ver-

3 Versuche mit dem Cosmic Ray Muon Detector

füßbarer Satelliten stark schwankten, wurden geografischer Breiten- und Längengrad mit Google Maps, Abstände mit dem „BayernViewer“ der Bayerischen Vermessungsverwaltung bestimmt. Diese Daten sind ebenfalls mit gewissen Ungenauigkeiten behaftet. Bei der Messung der Detektorkoordinaten (x,y,z) treten Ungenauigkeiten im Bereich von ≈ 10 cm auf. Außerdem wird in der Rekonstruktion ein punktförmiger Detektor angenommen. In der Realität hat er aber eine Ausdehnung von $30 \text{ cm} \times 25 \text{ cm}$.

Die Zeitmessung hat eine Auflösung von $t_{res} = 1,25 \text{ ns}$ innerhalb eines Detektorsets. Dadurch können bei der Rekonstruktion von Azimut und Höhe nur bestimmte Werte angenommen werden (siehe Abbildung 3.23). Die Zeitauflösung ist also direkt mit

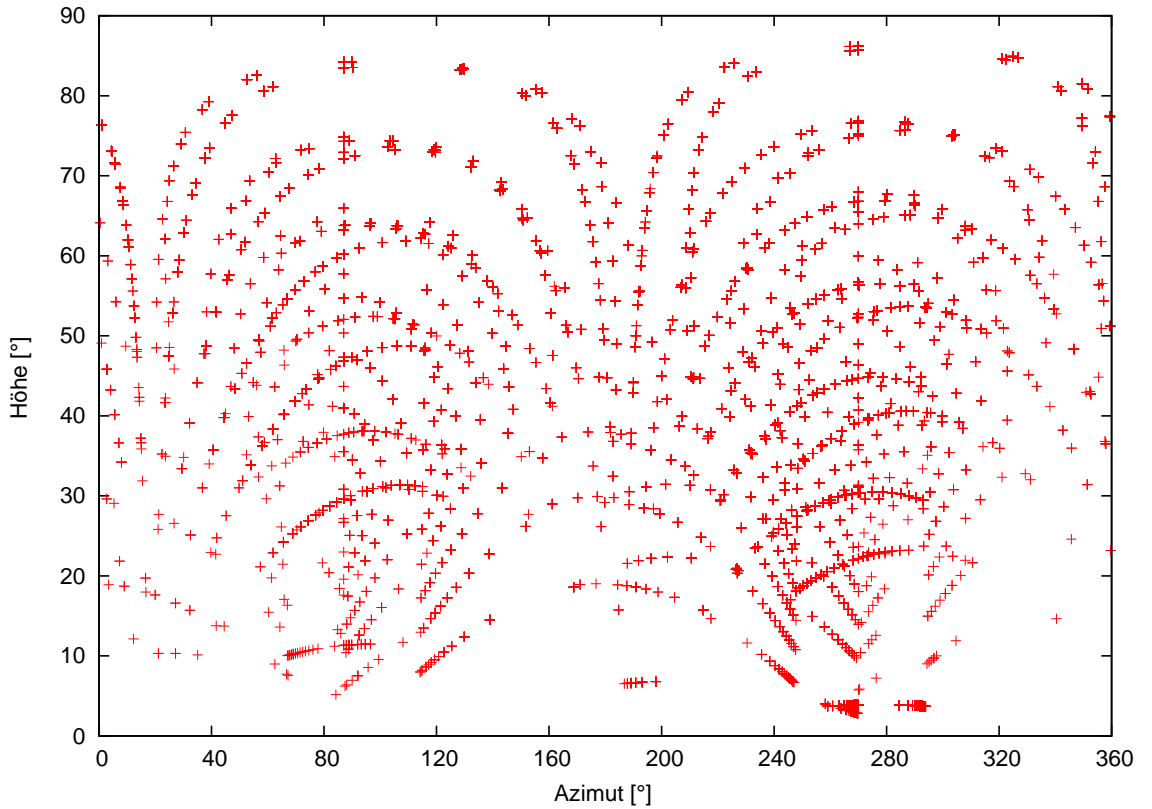


Abbildung 3.23: Rekonstruktion von Azimut und Höhe der 30001 gemessenen Schauer (GnuPlot)

der Winkelauflösung des Detektorsets verbunden (siehe Gleichung (3.8) und Abbildung 3.20):

$$\Delta\phi(d) = \frac{1}{\sqrt{1 - \left(\frac{t_{res} \cdot c}{d}\right)^2}} \cdot \frac{c \cdot t_{res}}{d}.$$

. Werden nun der minimale und maximale Detektorabstand von Set II aus Aufbau 2, mit welchem die Richtungsbestimmung durchgeführt wurde, eingesetzt und die Messfehler des Detektorabstands vernachlässigt, so ergeben sich als minimale und maximale

3 Versuche mit dem Cosmic Ray Muon Detector

Auflösung:

$$\Delta\phi_{min} = \Delta\phi(2,11 \text{ m}) \approx 10,3^\circ$$

$$\Delta\phi_{max} = \Delta\phi(5,85 \text{ m}) \approx 3,7^\circ.$$

Es ist ebenfalls wichtig, Datensätze zu verwenden, die ausschließlich ganze Tage (24 Stunden Aufzeichnung) enthalten. Andernfalls sind in den Auswertungen Vorzugsrichtungen (Bahnen) zu erkennen, welche durch die Rotation der Erde verursacht werden. Es wird stets nur ein Himmelsausschnitt betrachtet.

Die mit dem Java-Tool **anglet** ermittelten Daten unterliegen noch den Ungenauigkeiten der im Programm verwendeten Datentypen. Der für Fließkommazahlen verwendete Typ `double` hat eine Genauigkeit von 15 signifikanten Stellen.

3.5 Weitere Experimente

Neben den oben genannten Versuchen gibt es natürlich noch andere Möglichkeiten, Messungen mit dem CRMD-Set durchzuführen. In diesem Abschnitt werden drei Vorschläge für weitere Experimente kurz skizziert.

3.5.1 Abhängigkeit der Schauerrate von der Detektoranordnung

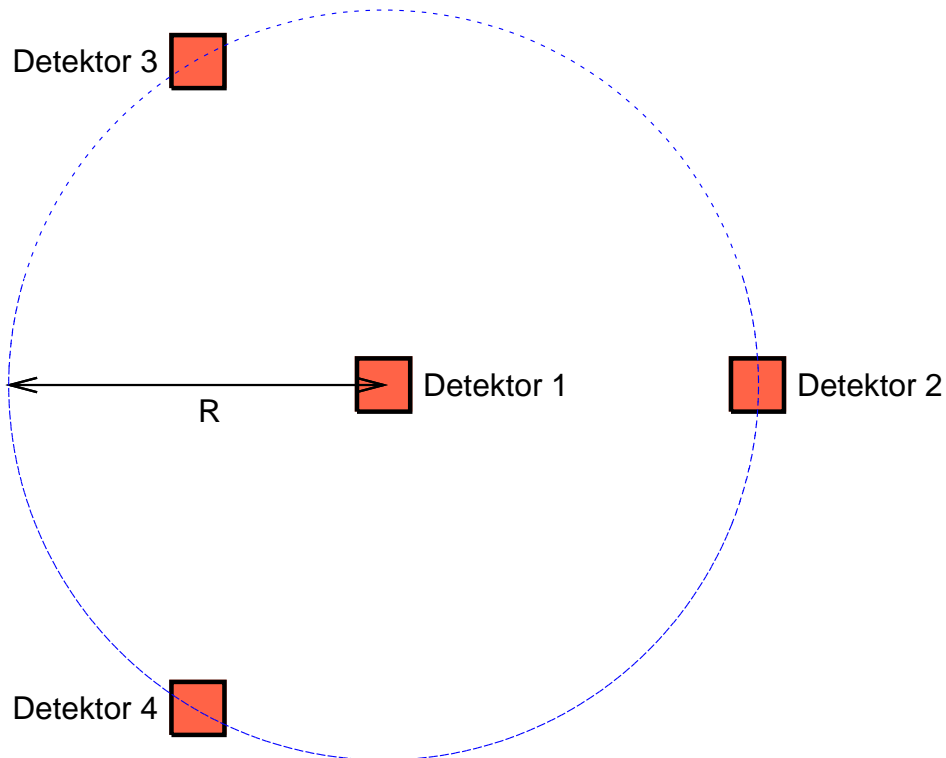


Abbildung 3.24: Detektoranordnung in Abhängigkeit von R (GnuPlot)

Bei diesem Versuch soll die Schauerrate bei verschiedenen Ausdehnungen R des Arrays (siehe Abbildung 3.24) gemessen werden. Dazu wird das Detektorset in Viererkoinzidenz betrieben und die Daten für Array-Radien zwischen 1 m und 15 m (Kabellänge) über einen bestimmten Zeitraum aufgenommen. Anschließend müssen daraus die Raten (Treffer pro Zeiteinheit) errechnet werden. Aus den Daten sollte nun ersichtlich sein, dass die Schauerrate bei steigender Arrayausdehnung abnimmt, was vor allem durch das zu großen Energien hin abfallende Energiespektrum der kosmischen Strahlung verursacht wird.

3.5.2 Geschwindigkeitsmessung

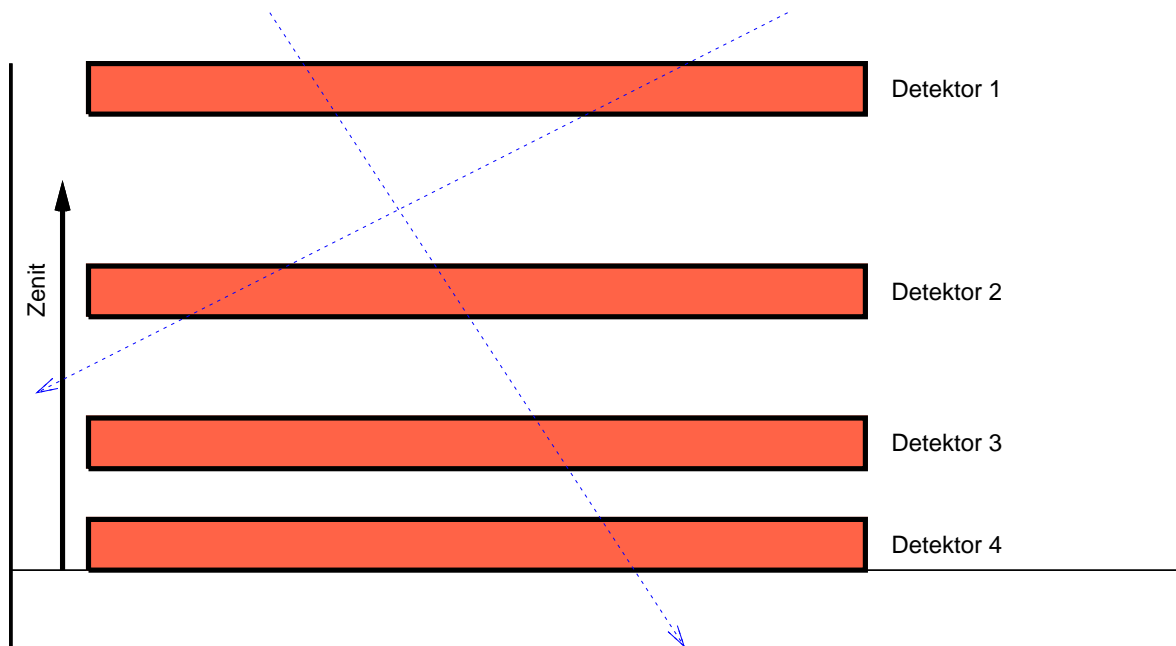


Abbildung 3.25: Möglicher Aufbau zur Messung der mittleren Myonengeschwindigkeit (GnuPlot)

Mit einem Versuchsaufbau aus drei oder vier Detektoren in vertikaler Anordnung (siehe Abbildung 3.25) lässt sich die Geschwindigkeit der Myonen messen. Dazu müssen sich die Detektoren in bekannten, nicht unbedingt gleich großen Abständen genau übereinander befinden. Bei der Wahl der Abstände muss die Zeitauflösung des Detektors von 1,25 ns berücksichtigt werden. Die einzelnen Kanäle sind je nach Anzahl der verwendeten Detektoren in Dreier- oder Viererkoinzidenz zu schalten. Mit Hilfe einer Monte-Carlo-Simulation muss die mittlere zurückgelegte Distanz zwischen den angesprochenen Detektoren abgeschätzt werden, da der Myonenfluss abhängig vom Einfallswinkel θ ist. Aus der so ermittelten Strecke und den Daten der dafür benötigten Zeiten kann nun die mittlere Geschwindigkeit der Myonen berechnet werden. Genauere Hinweise zur Versuchsdurchführung und Datenanalyse sind in [SL07] zu finden.

3.5.3 Vermessung der Größe des Mondschattens

Ein sehr interessanter Versuch zur kosmischen Strahlung mit einer Detektoranordnung, die sehr stark dem CRMD-Set ähnelt, wurde von Mitarbeitern der Sharif University of Technology in Teheran vor kurzem im Magazin „Astroparticle Physics“ veröffentlicht [SBS10]. In dieser wird der vom Mond verursachte Schatten in der kosmischen Strahlung untersucht. Dazu werden vier Detektoren quadratisch wie in Abbildung 3.15(a)

3 Versuche mit dem Cosmic Ray Muon Detector

mit einer Seitenlänge von 8,5 m angeordnet. Nach Rekonstruktion der Schauerrichtungen wie in Abschnitt 3.4 muss für jedes Ereignis auch die Zeit festgehalten werden, um daraus die Position des Mondes ermitteln zu können. Mit Hilfe der sphärischen Geometrie wird nun der Winkelabstand zwischen der Einfallrichtung des Schauer und der Mondposition berechnet und überprüft, ob dieser kleiner als der Winkelradius eines vorher gewählten „Fehler-Kreises“ ist. In [SBS10] werden in der Datenanalyse hierzu Winkelradien von $[0, 1; 0, 3; \dots; 6, 1]$ durchlaufen. Gleichzeitig wird je nach Größe des Fehler-Kreises eine Anzahl zufälliger Kreispositionen erzeugt. Diese dürfen einander nicht überlappen, daher sinkt bei Vergrößerung des „Fehler-Kreises“ die Anzahl der zufälligen Positionen. Nun werden ebenfalls die Winkelabstände zwischen dem Ereignis und den zufälligen „Fehler-Kreisen“ bestimmt. Um systematische Effekte wie Luftdruckschwankungen und Temperaturänderungen zu unterdrücken, werden jedoch nur die zufälligen Kreispositionen mit gleichen Zenit-Winkel θ ausgewählt (wegen $I \propto \cos^2 \theta$). Zur Auswertung wird die Anzahl der Ereignisse, die in den „Fehler-Kreis“ des Mondes fallen, mit der Anzahl der Ereignisse aus willkürlichen Richtungen (zufällige Positionen) verglichen. Werden die Daten weiter mit statistischen Methoden analysiert, wird ersichtlich, dass das Defizit der Treffer aus Richtung des Mondes signifikant ist. Über die Formulierung eines Modells zur erwarteten Ereignisanzahl aus den zufälligen Himmelsrichtungen in Abhängigkeit von der Größe des „Fehler-Kreises“ kann nun der Radius des Mondschattens bestimmt werden. Die in [SBS10] verwendeten Daten enthalten $1,7 \cdot 10^5$ Ereignisse bei einer Messdauer von 280 h. Dies entspricht einer Rate von ca. 10 Ereignissen/Minute. Die dabei verwendeten Szintillatoren hatten eine Größe von $100 \times 100 \times 2 \text{ cm}^3$ und waren somit deutlich größer als die CRMD-Szintillatoren.

4 Java-Werkzeuge zur Datennahme und Auswertung

Es gibt zwei Möglichkeiten die Messdaten des **CRMD** auszuwerten. Die erste Möglichkeit ist die Auswertung mit der Online-Plattform „Cosmic Ray eLab“, welche sehr übersichtlich und einfach zu bedienen ist. Sie hat jedoch den Nachteil, dass viele Prozesse bei der Datenbearbeitung und Selektion im Verborgenen bleiben. Als Alternative bleibt nur die Daten selbst auszuwerten. Dies liefert den SchülerInnen gleichzeitig auch einen guten Einblick in die Arbeitswelt eines Physikers. Aufgrund der großen Datenmengen sind gängige Tabellenkalkulationsprogramme für die Auswertung ungeeignet. Daher musste eine Reihe von Java-Programmen zur Datennahme, Kalibration, Datenbearbeitung und Auswertung im Rahmen dieser Arbeit selbst geschrieben werden. Diese werden hier mit ihren Funktionen und Anforderungen vorgestellt. Jedoch muss vorweg gesagt werden, dass sich die Programmierkenntnisse des Verfassers auf einem niedrigen Niveau befinden. Vor allem das „exception handling“ ist ein Schwachpunkt. Aber die Programme erfüllen ihre Zwecke und dürfen gerne jederzeit verbessert und erweitert werden (Quelltexte befinden sich im Anhang).

Da Java eine plattformunabhängige Programmiersprache ist, sind die Programme auf allen Betriebssystemen ausführbar. Jedoch muss vorher die entsprechende „Java Virtual Machine“ installiert werden. Diese steht als „Java Runtime Environment“ (**JRE**) zum Download bereit (<http://www.java.com/de/download/>). Zusätzlich muss die Erweiterung **RXTX** installiert werden, die die Programmbibliotheken zum Schnittstellenzugriff (Paket gnu.io.*) enthält. Auch **RXTX** ist kostenlos als Download erhältlich (<http://rxtx.qbang.org/pub/rxtx/rxtx-2.1-7-bins-r2.zip>). Es müssen zwei Dateien im Java-Verzeichnis platziert werden:

- Windows: „RXTXcomm.jar“ in `\jre\lib\ext` und „rxtxSerial.dll“ in `\jre\bin`.
- Linux: „RXTXcomm.jar“ in `/jre/lib/ext` und „librxtxSerial.so“ in `/jre/lib/[machine type]` (machine type meist i386).

Nun können alle hier aufgeführten Programme verwendet werden. Java-Programme haben die Dateiendung „.class“ („beispiel.class“). Sie werden über die Kommandozeile (Eingabeaufforderung bei Windows, Terminal bei Linux) mit dem Befehl **java** aufgerufen („java beispiel“). Dabei muss sich die Programmdatei in dem Verzeichnis befinden, in welchem der Aufruf stattfindet oder ein relativer Pfad angegeben werden. Zu beachten ist außerdem, dass Fließkommazahlen stets mit '.' statt ',' getrennt werden müssen.

4.1 Kalibration mit autocal

Die Kalibration des Threshold-Level ist sehr zeitaufwändig. Die Schwellen müssen per Board-Befehl gesetzt werden, dann muss ca. fünf Minuten gewartet werden, bis die Zähler ausgelesen werden können. Anschließend beginnt der Prozess wieder von vorne. Um sich diese Zeit zu sparen, wurde der Prozess durch das Programm **autocal** automatisiert.

Das Programm **autocal** erfordert 3 Eingabeparameter: den COM-Port, den das DAQ-Board benutzt, das Aufzeichnungsintervall in Minuten und die Ausgabedatei.

Aufruf: java autocal <COM#> <interval [min]> <outputfile>

Das Programm durchläuft die Spannungswerte von 0 bis 600 mV in 25 mV-Schritten. In der Ausgabedatei können anschließend die Schwellen und die zugehörigen Zählerwerte (HEX-Format) eingesehen werden. Dabei entspricht Zähler „DS#“ (0, 1, 2, 3) Kanal #. „DS4“ steht für die Anzahl der Trigger, „DS5“ für die Boardzeit.

Beispiel: java autocal COM3 2.5 output.txt

Im Beispiel wird alle 2,5 min der Zähler ausgelesen und in die Datei „output.txt“ geschrieben. Das DAQ-Board wird an Schnittstelle COM3 betrieben.

4.2 Datennahme und Bearbeitung

Zur Datennahme wird hier ein selbst erstelltes (plattformunabhängiges) Terminal-Programm vorgestellt. Aufgrund der Größe der Datendateien kann zur Ansicht kein gängiger Texteditor verwendet werden. Hierfür wurde ebenfalls eine Lösung entwickelt. Zusätzlich werden zwei Programme zur Fehlerkorrektur der Daten dargestellt.

4.2.1 getdata

Zwar bietet Windows das Hyperterminal zur Kommunikation mit Schnittstellen an, dieses arbeitet jedoch sehr unzuverlässig. Daher wurde das Java-Programm **getdata** entwickelt, das diese Aufgabe deutlich zuverlässiger erledigt. Als Grundlage dafür wurde der Quelltext des Programms „comframe“ [jav] verwendet.

Aufruf: `java getdata <COM#> <outputfile>`

Das Programm öffnet ein Fenster, in welchem nun die Board-Befehle zur Steuerung eingegeben werden können.

Beispiel: `java getdata COM3 output.txt`

Im Beispiel werden sämtliche Daten, die über die Schnittstelle COM3 gesendet werden, in der Datei „output.txt“ gespeichert.

Ausgabe: Datei im Rohdaten-Format (Abschnitt 4.4.1).

4.2.2 linecheck

Die Dateigrößen der aufgezeichneten Daten sind meist zu groß (GigaByte-Bereich), um mit einem Texteditor geöffnet zu werden. Mit dem Programm **linecheck** ist es trotzdem möglich, einen Blick in die Daten zu werfen.

Eingabe: Datei im Rohdaten-Format (Abschnitt 4.4.1)

Es kann auf verschiedene Arten aufgerufen werden. Aufruf 1 gibt einen gewissen Ab-

schnitt auf dem Bildschirm aus, wobei Anfangs- und Endzeile eingegeben werden können.

Aufruf 1: `java linecheck <inputfile> <startline> <endline>`

Beispiel 1: `java linecheck input.txt 20 40`

Im Beispiel werden so die Zeilen 20 bis 40 der Datei „input.txt“ auf dem Bildschirm ausgegeben. Dabei ist zu beachten, dass die Zählung der Zeilen bei Null beginnt. Soll ein gewisser Abschnitt aus einer Datei ausgeschnitten werden, muss dieser dem Programm in Form der entsprechenden Anfangs- und Endzeile mitgeteilt werden.

Aufruf 2: `java linecheck <inputfile> <startline> <endline> <outputfile>`

Beispiel 2: `java linecheck input.txt 20 40 output.txt`

Im Beispiel werden die Zeilen 20 bis 40 der Datei „input.txt“ in die Datei „output.txt“ geschrieben.

Ausgabe Aufruf 2: Datei im Rohdaten-Format (Abschnitt 4.4.1) enthält ausgewählte Zeilen

Das Programm ist auch in der Lage, die Datenzeilen ausgewählter Zeitpunkte (hhmmss ddmmyy) auszugeben, wenn beim Aufruf die folgenden Parametern übergeben werden:

Aufruf 3: `java linecheck <inputfile> <hhmmss> <ddmmyy> t`

Beispiel 3: `java linecheck input.txt 181920 080910 t`

Im Beispiel werden auf dem Bildschirm alle Datenzeilen der Uhrzeit 18:19:20 des 8. September 2010 ausgegeben.

4.2.3 fill

Da es bei der Aufzeichnung von Daten öfters zu „abgeschnittenen“ Zeilen kam, wurde das Tool **fill** entworfen.

Aufruf: `java fill <inputfile> <outputfile>`

Das Programm vervollständigt die Zeitinformationen, falls genügend Reliktinformationen zur Rekonstruktion vorhanden sind. Ungültige Zeilen werden aus den Daten entfernt.

Eingabe: Datei im Rohdaten-Format (Abschnitt 4.4.1) mit korrupten Zeilen

In die Ausgabedatei werden die verbesserten Daten der Eingabedatei geschrieben. Außerdem wird eine weitere Datei erstellt, welche die aussortierten Zeilen enthält. So kann überprüft werden, ob das Programm nur zerstörte Zeilen entfernt hat.

Beispiel: `java fill input.txt output.txt`

Im Beispiel wird die Datei „input.txt“ gefiltert. Korrekte oder korrigierte Daten werden in „output.txt“ geschrieben, defekte Zeilen landen in „recycling output.txt“.

Ausgabe: zwei Dateien im Rohdaten-Format (Abschnitt 4.4.1) mit korrigierten und entfernten Zeilen

4.2.4 correct

Beim Hochladen der Dateien auf die Online-Plattform „Cosmic Ray eLab“ trat oft das Problem auf, dass die Datumsinformation kurz nach Mitternacht zu spät aktualisiert wurde. Falsche Datumsinformationen haben zur Folge, dass das Online-Skript nur die Daten mit dem fehlerhaften Datum ausliest und den Rest entfernt. Das Programm **correct** korrigiert vorliegende Datumsfehler in den Daten.

Aufruf: `java correct <inputfile> <startdate: ddmmyy> <outputfile>`

Eingabe: Datei im Rohdaten-Format (Abschnitt 4.4.1) mit falscher Datumsinformation

Beispiel: `java correct input.txt 080910 output.txt`

Im Beispiel werden die Daten der Datei „input.txt“, deren Aufzeichnung am 8. September 2010 begann, verbessert und in die Datei „output.txt“ geschrieben.

Ausgabe: Datei im Rohdaten-Format (Abschnitt 4.4.1) mit korrigierter Datumsinformation

4.2.5 sort

Vor der Richtungsrekonstruktion der Luftschauer müssen die Daten jedes Tages chronologisch geordnet werden. Dies geschieht mit Hilfe des „Bubble-Sort“-Verfahrens durch das Programm **sort**.

Eingabe: Datei im readout-Format (Abschnitt 4.4.2)

Aufruf: `java sort <inputfile>`

Die Daten werden aus der Eingabedatei gelesen, nach der Zeit sortiert und anschließend in die Ausgabedatei geschrieben.

Beispiel: `java sort input.txt`

Die geordneten Daten der Datei „input.txt“ finden sich in der Datei „sortedinput.txt“ wieder.

Ausgabe: Datei im readout-Format (Abschnitt 4.4.2) mit chronologisch sortierten Datenzeilen

4.3 Auswertung

Die Auswertung ist aufgrund der großen Datenmenge ohne Computer undenkbar. In diesem Abschnitt werden Java-Tools vorgestellt, welche die (meist rechenaufwändige) Datenauswertung erledigen. Die Darstellung der Resultate erfolgte in dieser Arbeit stets mit **GnuPlot**.

4.3.1 readout

Die Zeitinformationen in den Rohdaten sind größtenteils im Hex-Format (siehe Abschnitt 4.4.1) und somit schlecht lesbar.

Aufruf: `java readout <inputfile> <outputfile>`

Um die Daten lesen und weiterverarbeiten zu können, müssen die Hex-Daten umgerechnet werden (siehe Abschnitt 2.2.4). Dies wird vom Programm **readout** erledigt.

Eingabe: Datei im Rohdaten-Format (Abschnitt 4.4.1)

Als Parameter werden die Eingabedatei und die Ausgabedatei, in welcher die lesbaren Daten gespeichert werden, übergeben. Es werden insgesamt fünf Ausgabedateien erstellt, da zusätzlich zur übersetzten Datei für die Ausgabe jedes Kanals eine eigene Datei angelegt wird.

Beispiel: `java readout input.txt output.txt`

Im Beispiel werden die Rohdaten als „input.txt“ eingelesen und übersetzt in der Datei „output.txt“ gespeichert. Die Daten der einzelnen Kanäle können in den Dateien „outputCH0.txt, outputCH1.txt, outputCH2.txt, outputCH3.txt“ eingesehen werden.

Ausgabe: fünf Dateien im readout-Format (Abschnitt 4.4.2) (eine Sammeldatei, vier Kanaldateien)

4.3.2 gnutot

Zur Überprüfung der Kalibration ist es nützlich, sich die **TOT**-Verteilung der einzelnen Kanäle anzusehen.

Aufruf: `java gnutot <inputfile> <binwidth [ps]> <limit [ps]> <outputfile>`

Um ein Histogramm der **TOT**-Zeiten zu erstellen, wird das Java-Tool **gnutot** benutzt.

Eingabe: Datei im readout-Format (Abschnitt 4.4.2)

Als Parameter werden Ein- und Ausgabedatei, die Bin-Breite in ps und die Obergrenze in ps übergeben.

Beispiel: `java gnutot input.txt 1250 50000 output.txt`

Im Beispiel wird das Intervall $[0; 50000]$ in die Teilintervalle $]0; 1250]$, $]1250; 2500]$, \dots , $]48750; 50000]$ zerlegt und die Daten aus „input.txt“ diesen Intervallen zugeordnet. Die Anzahl der Treffer der einzelnen Intervalle wird in Datei „output.txt“ gespeichert und kann nun als Histogramm geplottet werden.

Ausgabe: Datei im gnutot-Format (Abschnitt 4.4.3)

4.3.3 rate

Das Ziel des Versuchs in Abschnitt 3.2 ist die zeitliche Erfassung des Myonenflusses.

Aufruf: `java rate <inputfile> <binwidth [s]> <detectorarea [cm2]> <outputfile>`

Zur Darstellung des Myonenflusses müssen die Daten mit dem Programm **rate** ausgewertet werden.

Eingabe: Datei im readout-Format (Abschnitt 4.4.2)

Als Parameter werden Ein- und Ausgabedatei, die Bin-Breite in s und die Detektorfläche in cm² übergeben.

Beispiel: `java rate input.txt 600 750 output.txt`

Im Beispiel wird die Myonenrate pro Minute und m^2 aus den Daten „input.txt“ für eine Bin-Breite von $600 \text{ s} = 10 \text{ min}$ und einer Detektorfläche von 750 cm^2 ermittelt und in „output.txt“ gespeichert.

Ausgabe: Datei im rate-Format (Abschnitt 4.4.4)

4.3.4 anglet

Die Richtung von Luftschauern spielt eine wichtige Rolle in Abschnitt 3.4. Die Umrechnung aus den Daten besteht aus mehreren Transformationen.

Aufruf: `java anglet <inputfile> <outputfile>`

Die Richtungsrekonstruktion der Luftschauer wird vom Programm **anglet** durchgeführt.

Eingabe: Datei im readout-Format (Abschnitt 4.4.2), Datei „config.txt“ im config-Format (Abschnitt 4.4.5) mit den Detektorkoordinaten

Es erwartet als Parameter für den Programmaufruf lediglich eine Ein- und Ausgabedatei. Zusätzlich muss sich im gleichen Verzeichnis die Datei „config.txt“ mit den Detektorkoordinaten befinden.

Beispiel: `java anglet input.txt output.txt`

In die Ausgabedatei werden sämtliche Richtungsinformationen der jeweiligen Schauer geschrieben.

Ausgabe: Datei im anglet-Format (Abschnitt 4.4.6)

4.3.5 histo

Zur Darstellung der Verteilung der Schauerrichtungen müssen die Daten gebinnt werden, also die Anzahl pro Intervall ermittelt werden.

Aufruf: `java histo <xtics> <xmin> <xmax> <ytics> <ymin> <ymax> <input-file> <outputfile>`

Die Schauerrichtungen werden vom Programm **anglet** geliefert. Für das Binning wird das Programm **histo** verwendet.

Eingabe: Datei im anglet-Format (Abschnitt 4.4.6)

Dabei gibt xmin das Minimum der x-Werte, xmax das Maximum de x-Wert und xtics die Intervallbreite an (y analog). Die pro Intervall gezählten Werte der Eingabedatei werden mit den entsprechenden x- und y-Werten in die Ausgabedatei geschrieben.

Beispiel: `java histo 1 0 24 10 0 90 input.txt output.txt`

Im Beispiel wird das x-Intervall $[0; 24]$ in die Teilintervalle $[0; 1[$, $[1; 2[$, \dots , $[23; 24[$ und das y-Intervall $[0; 90]$ in die Teilintervalle $[0; 10[$, $[10; 20[$, \dots , $[80; 90[$ zerlegt. Die Daten aus „input.txt“ werden den Intervallen zugeordnet und die Anzahl der Treffer des jeweiligen x-y-Intervalls in die Datei „output.txt“ geschrieben. Die Daten können nun in einem 3D-Histogramm dargestellt werden.

Ausgabe: Datei im histo-Format (Abschnitt 4.4.7)

4.4 Datenformate

In diesem Abschnitt werden die verschiedenen Formate der Daten vorgestellt, die von den in Kapitel 4 angegebenen Programmen verwendet werden. Jedes Format erfüllt dabei einen speziellen Zweck.

4.4.1 Rohdaten-Format

```
3774BBFE 80 00 00 00 00 00 00 3C 00 370A867C 000021.022 030610 A 08 0 +0072
3774BBFF 00 00 23 00 00 00 00 00 00 370A867C 000021.022 030610 A 08 0 +0072
3774BBFF 00 00 00 31 00 00 00 00 00 370A867C 000021.022 030610 A 08 0 +0072
3774BC00 00 00 00 00 00 00 00 00 21 370A867C 000021.022 030610 A 08 0 +0072
```

Das Rohdaten-Format ist das Ausgabeformat des DAQ-Boards. Die Daten, die durch die Terminalsoftware (**getdata**, Hyperterminal, etc.) übertragen werden, haben diese Form. Jede Zeile besteht aus 16 Wörtern. Eine genauere Beschreibung der einzelnen Wörter ist der Anleitung [RJPB10] zu entnehmen.

4.4.2 readout-Format

```
21278420595000 46250 030610 3
21278420603750 17500 030610 1
```

Daten im Rohdaten-Format können mit dem Programm **readout** ausgelesen werden. Das Format der so bearbeiteten Daten ist das readout-Format. Es hat den Vorteil, dass es leichter lesbar ist und zugleich weniger Speicherplatz benötigt (Datenzeilen in Abschnitt 4.4.1 entsprechen denen in Abschnitt 4.4.2). Eine Zeile besteht aus vier Wörtern. Das erste Wort ist die Zeit seit Mitternacht in ps. Wort zwei steht für die Zeit oberhalb der Schwelle (**TOT**) ebenfalls in ps. Das dritte Wort repräsentiert das Datum des Signals. Der Kanal, der das Signal ausgelöst hat, wird im vierten Wort festgehalten (wichtig: Zählung der Kanäle beginnt bei 0).

4.4.3 gnutot-Format

```
0      2500 17
2500 5000 52
5000 7500 230
7500 10000 356
```

Das Format wurde geschaffen, um die **TOT**-Werte zu „binnen“, damit sie in einem Histogramm mit Hilfe eines Plot-Programms dargestellt werden können. Dies dient vor allem der Überprüfung der Kalibration. Das gnutot-Format enthält drei Informationen. Die Worte eins und zwei stehen für die Unter- und Obergrenze des betrachteten Intervalls (bins) in μs . Wort drei stellt die Anzahl der Detektortreffer mit **TOT** im entsprechenden Intervall zurück.

4.4.4 rate-Format

```
47915207628486250 7893
47930207628486250 7360
47945207628486250 8373
47960207628486250 7786
```

Mit dem Programm **rate** ist es möglich aus den Daten die Zählrate zu bestimmen. Das Ausgabeformat des Programms besteht aus zwei Wörtern. Wort eins ist die Zeit seit Mitternacht in μs . Die Zählrate in $\text{min}^{-1}\text{m}^{-2}$ ist in Wort zwei angegeben. Der zeitliche Verlauf kann nun mit einem Plot-Programm dargestellt werden.

4.4.5 config-Format

```
0 0 0 0 -5.15 0 -2 -4.5 0.15 -2.25 0.25 0.15 49.579632 11.030627
```

Zur Richtungsrekonstruktion der Luftschaue mit dem Programm **anglet** werden die Detektorkoordinaten benötigt. Da einer Übergabe dieser mittels Kommandozeile eine zu große Fehlerquelle darstellt, werden sie über die Datei „config.txt“ eingelesen, welche sich im gleichen Verzeichnis wie die Klassendatei „anglet.class“ befinden muss. Die

Worte eins bis drei stehen für die x -, y - und z -Koordinate des ersten Detektors (x -Achse Richtung Osten, y -Achse Richtung Norden und z -Achse Richtung Zenit). Analog finden sich in den Worten vier bis zwölf die Koordinaten der Detektoren zwei, drei und vier. Breiten- und Längengrad müssen an 13. und 14. Stelle übergeben werden.

4.4.6 anglet-Format

```
58.55335545933653 80.57608074524634 23.754314506193538 44.07870755081148
90.17508890210996 83.53335948543918 0.05690266020401924 49.17301871247774
89.82379728385555 84.2467523094829 0.16161010706081136 49.22439018728468
56.10373585944282 82.5295507505336 0.04512315923905885 45.058821820758986
```

Das Programm **anglet** gibt seine für einen Luftschauer errechneten Werte in einer Zeile aus. Jede Zeile besteht aus vier Wörtern, die die Richtungsinformationen enthalten. Das erste Wort gibt den Azimutwinkel a , das zweite die Höhe h des Schauer in $^\circ$ an. Die Worte drei und vier stehen für die Rektaszension α in h und die Deklination δ in $^\circ$. Das Format kann auch leere Zeilen erhalten, falls die Rekonstruktion der Richtung fehlschlägt. Dies kann aufgrund von zufälligen Koinzidenzen der Fall sein.

4.4.7 histo-Format

```
0.50 14.00 7
0.50 18.00 18
0.50 22.00 16
0.50 26.00 32
```

Um die rekonstruierten Luftschauerrichtungen in einer Himmelskarte (Histogramm) darzustellen, müssen die Daten mit dem Programm **histo** gebinnt werden. Jedes dreidimensionale Bin nimmt eine Zeile ein. Wort eins steht für den rechten Rand des Rektaszensionintervalls. Wort zwei gibt dementsprechend den Randwert für die Deklination an. Die Anzahl der im Intervall befindlichen Schauer findet sich in Wort drei wieder. Diese Daten können mit einem Plot-Programm dargestellt werden.

5 Array-Studie

Ein Ziel des Schulprojekts, das in Kapitel 6 genauer erläutert wird, ist es, ausgedehnte Teilchen-Luftschauer mit einem Array aus CRMD-Sets zu messen. Diese sollen an Schulen in Erlangen (Ausdehnung des Stadtgebiets: ungefähr $7 \text{ km} \times 7 \text{ km}$) betrieben werden. Die Positionen und Arten der Schulen sind in Abschnitt 5.1 aufgeführt. Eine erste Abschätzung der zu erwartenden Ereignisrate der Detektoren wird in Abschnitt 5.2 erläutert.

5.1 Positionen der Schulen in Erlangen

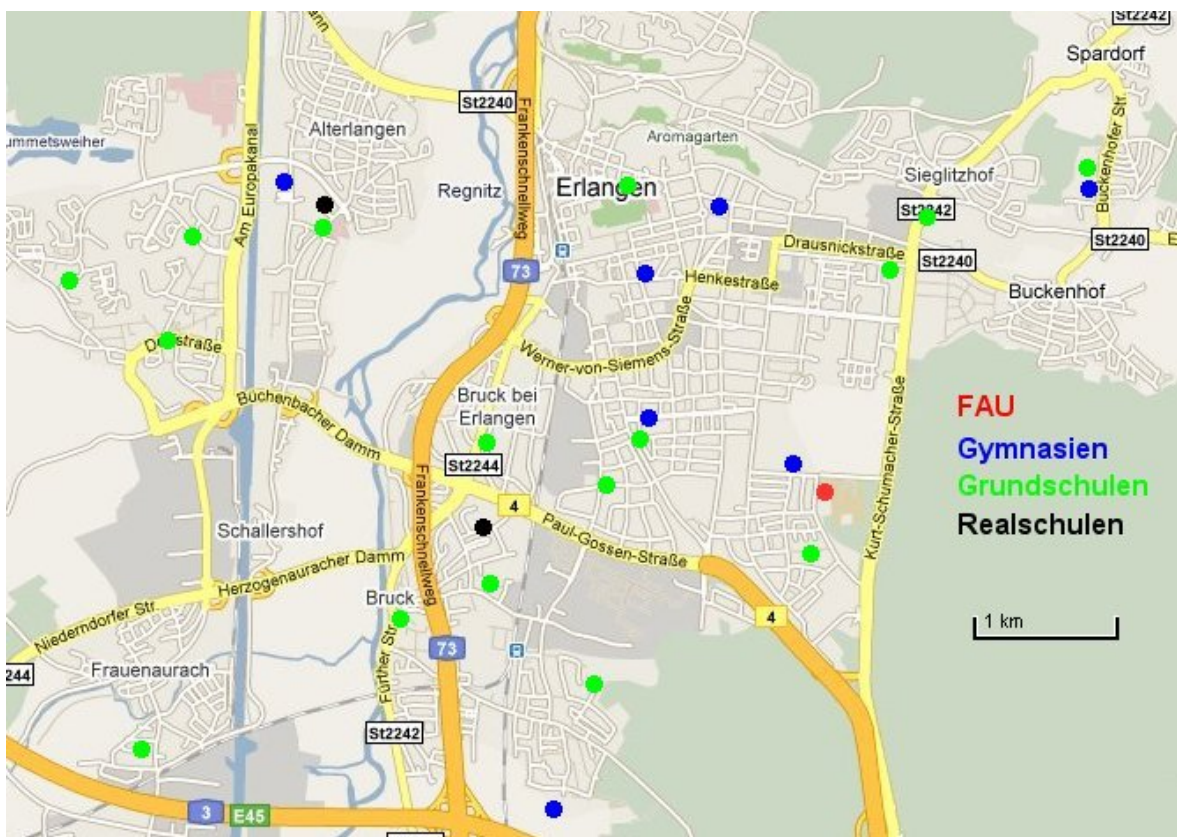


Abbildung 5.1: Schulen in Erlangen (GoogleMaps)

Tabelle 5.1: Positionen der Erlanger Gymnasien ermittelt mit GoogleMaps

Gymnasium	Adresse	PLZ	Breite [°]	Länge [°]
Fridericianum	Sebaldusstraße 37	91058	49,582199	11,025605
Ohm	Am Röthelheim 6	91052	49,584954	11,010446
Christian-Ernst	Langemarckplatz 2	91054	49,594270	11,010306
Emmy-Noether	Noetherstraße 49	91058	49,560895	11,001010
Albert Schweitzer	Dompfaffstraße 111	91056	49,599868	10,976070
Marie-Therese	Schillerstraße 12	91054	49,598933	11,017098
Emil-von-Behring	Buckenhofer Straße 5	91080	49,600619	11,053372

In Erlangen gibt es sieben Gymnasien (blaue Punkte in Abbildung 5.1), die über das gesamte Stadtgebiet verteilt sind. Das SKYVIEW-Array in Wuppertal verwendet zusätzlich zu den Gymnasien auch Grund-, Haupt- und Realschulen als Detektorstandorte. In Abbildung 5.1 sind die Grund- und Realschulen ebenfalls eingezeichnet (grüne und schwarze Punkte). Die Lage des Physikalischen Instituts (ECAP) ist mit einem roten Punkt gekennzeichnet.

In Tabelle 5.1 sind alle Gymnasien in Erlangen mit Adresse, geographischer Breite und geographischer Länge aufgeführt. Die Positionsangaben wurden mit GoogleMaps ermittelt und beziehen sich jeweils auf das Hauptgebäude der Schule.

5.2 Simulationsuntersuchungen zur Messung der kosmischen Strahlung mit Szintillationsdetektoren

Zur Abschätzung der Myonenzählrate eines CRMD-Arrays, dessen Detektoren an verschiedenen Schulen Erlangens platziert werden sollen, führte Stefan Fryska mehrere CORSIKA-Simulationen im Rahmen seiner Bachelorarbeit [Fry10] durch. Es handelt sich dabei um eine untere Abschätzung der zu erwartenden Ereignisse, bei der mehrere Annahmen getroffen wurden, die den ungünstigsten Fall (kleinste Ereignisrate) repräsentieren:

- Es werden nur Schauer bei bestimmten Energien des Spektrums simuliert.
- Das Primärteilchen ist ein Proton.
- Das Primärteilchen tritt senkrecht in die Atmosphäre ein.
- Das Primärteilchen tritt direkt über dem ECAP in die Atmosphäre ein.
- Die erste Interaktion findet in 25 km Höhe statt.

Außerdem musste die Anzahl der simulierten Teilchen begrenzt werden, um die Datenmenge und Rechenzeit im Rahmen zu halten. Dazu wurde die elektromagnetische Komponente des Schauers weitgehend ausgeblendet.

Ein Vorteil des Standorts Erlangen liegt in der Höhe von 300 m über dem Meeresspiegel. Die Simulationen ergaben, dass ab einer Energie von $E = 10^{17}$ eV das Schauernmaximum (größte laterale Ausdehnung der Sekundärteilchen-Verteilung) in eben dieser Höhe liegt und der Ort somit zur Untersuchung hochenergetischer Schauer geeignet ist.

Das physikalische Institut wurde als Schauerzentrum für die Simulation gewählt und orerst nur die Ereignisse an den drei dem ECAP nächstgelegenen Gymnasien berücksichtigt:

- Gymnasium Friedericianum (0,3 km zum ECAP)
- Ohm Gymnasium (1,4 km zum ECAP)
- Christian-Ernst Gymnasium (1,9 km zum ECAP)

Tabelle 5.2: Zählraten $[1/m^2 \cdot a]$ an Stelle des ECAP und der drei nächstgelegenen Schulen

Energie	ECAP	Friedricianum	Ohm	Christian-Ernst
E_1	$(4, 6 \pm 0, 9) \cdot 10^{-3}$	$(6, 4 \pm 2, 7) \cdot 10^{-5}$	$(4, 0 \pm 0, 8) \cdot 10^{-5}$	$(4, 0 \pm 0, 8) \cdot 10^{-5}$
E_2	$(6, 7 \pm 6, 0) \cdot 10^{-3}$	$(5, 0 \pm 4, 3) \cdot 10^{-5}$	$(3, 2 \pm 1, 9) \cdot 10^{-5}$	$(3, 8 \pm 1, 6) \cdot 10^{-5}$
E_3	$0, 2 \pm 0, 16$	$(5, 6 \pm 3, 2) \cdot 10^{-4}$	$(5, 5 \pm 1, 7) \cdot 10^{-4}$	0

Anschließend wurden drei Schauer, induziert durch Primärteilchen mit den Energien $E_1 = 4 \cdot 10^{18}$ eV, $E_2 = 1 \cdot 10^{19}$ eV und $E_3 = 1 \cdot 10^{20}$ eV, simuliert. Für die CRMD-Sets wurde jeweils eine quadratische Anordnung (3 m Seitenlänge) angenommen. Per Zufallsgenerator wurden 1000 Positionen innerhalb eines Radius von 2 km erzeugt und ihre simulierte Myonenrate in der Nähe der Schulabstände ausgelesen und gemittelt. Um daraus die Ereignisrate an den Standorten in $\frac{1}{m^2 \cdot a}$ zu erhalten, sind diese Zählraten noch mit den Auftrittswahrscheinlichkeiten der Primärteilchen multipliziert worden. Die daraus resultierenden Raten in $\frac{1}{m^2 \cdot a}$ sind in Tabelle 5.2 festgehalten. Dabei sind diese jeweils auf ein ganzes CRMD-Set (also vier Detektoren) zu beziehen.

Um eine bessere Vorstellung von den Werten zu vermitteln, wird ein Beispiel mit Hilfe der Binomialverteilung (Gleichung (5.1)) und den Laplaceschen Annahmen durchgespielt. Dabei ist p die Trefferwahrscheinlichkeit ($q = 1 - p$), n die Anzahl der Versuche und k die Anzahl der Treffer.

$$B(k|p, n) = \binom{n}{k} p^k q^{n-k} \quad (5.1)$$

Es wird der hypothetische Quadratmeter Atmosphäre betrachtet, der sich genau über dem ECAP befindet. Würde nun ein Teilchen pro Jahr an diesem Standort erwartet werden, dann läge die Wahrscheinlichkeit dafür, das Teilchen an einem bestimmten Tag an diesem Ort zu messen, bei $p = 1/365$. Wird nun von einer Messdauer von 90 Tagen ausgegangen, so ist die Wahrscheinlichkeit P , in diesem Zeitraum mindestens ein Teilchen an besagtem Ort zu messen:

$$P = 1 - \binom{90}{0} \left(\frac{1}{365}\right)^0 \left(\frac{364}{365}\right)^{90} = 0,219 = 21,9\%$$

Diese Rechnung wurde nun analog für die in Tabelle 5.2 aufgeführten Werte durchgeführt. Dabei wurde ebenfalls von einer plausiblen Messdauer von 90 Tagen ausgegangen. Die Ergebnisse sind in Tabelle 5.2 festgehalten.

Tabelle 5.3: Wahrscheinlichkeiten für mindestens ein Ereignis in 90 Tagen

Energie	ECAP	Friedricianum	Ohm	Christian-Ernst
E_1	1,13‰	0,02‰	0,01‰	0,01‰
E_2	0,02‰	0,01‰	0,01‰	0,01‰
E_3	48,13‰	0,14‰	0,14‰	0‰

Die Wahrscheinlichkeiten, mindestens ein Ereignis in 90 Tagen zu erhalten, (Tabelle 5.2) sind sehr gering. Die größte Wahrscheinlichkeit, ein Teilchen zu messen, liegt bei 4,81% am Standort des ECAP und einer simulierten Primärteilchenenergie $E_3 = 1 \cdot 10^{20}$ eV. Es muss jedoch stets beachtet werden, dass es sich hierbei um eine untere Abschätzung handelt, welche vor allem durch die betrachteten hohen Primärteilchenenergien (welche entsprechend selten vorkommen) eingeschränkt wird. Die Auswahl der extrem hohen Primärteilchenenergien für die Simulation war wiederum durch die Absicht begründet, Teilchenschauer mit einer „Myonenausleuchtung“ mehrerer Erlanger Detektorstandorte zu finden, mit denen eine Koinzidenzmessung innerhalb des Arrays möglich wäre.

Um verlässliche Werte für die zu erwartende Ereignisrate zu erhalten, sind umfangreichere Simulationen nötig. Diese sollten das gesamte Energiespektrum für die Primärteilchen abdecken. In den hier vorgestellten Berechnungen wurde als Primärteilchen stets ein Proton verwendet. Jedoch sollten in einer genaueren Simulation sämtliche Atomkerne in ihren natürlichen Häufigkeiten auftreten. Auch die hier getroffenen Annahmen zu Ort des Auftreffens auf die Atmosphäre, Höhe der ersten Interaktion und Einfallswinkel sollten durch Parameter ersetzt werden, die sich stärker an der Realität orientieren.

6 Wissenschaftspropädeutisches Seminar zur kosmischen Strahlung

Nachdem die Funktionsweise des CRMD erklärt und die damit möglichen Experimente in den vorhergehenden Kapiteln vorgestellt wurden, werden hier Möglichkeiten zur Durchführung eines W-Seminars zur kosmischen Strahlung aufgezeigt. Zunächst sollen in Abschnitt 6.1 die Vorgaben des bayerischen Kultusministeriums zum W-Seminar erläutert werden. Anschließend werden in Abschnitt 6.2 Wege zur Umsetzung dieser in die Praxis aufgezeigt. Eine mögliche Abfolge des Lehrplans wird in Abschnitt 6.3 vorgestellt. Abschnitt 6.4 bietet einen kleinen Überblick der Literatur zum Thema kosmische Strahlung.

6.1 Vorgaben

In diesem Abschnitt werden die Vorgaben des bayerischen Kultusministeriums zum „Wissenschaftspropädeutischen Seminar“ erläutert. Sämtliche Informationen sind der Broschüre „Die Seminare in der gymnasialen Oberstufe“ [MR08] entnommen, welches vom „bayerischen Staatsinstitut für Schulqualität und Bildungsforschung“ (**ISB**) herausgegeben wird.

6.1.1 Ziel: Kompetenz in wissenschaftlichem Arbeiten

Eine Studie des ISB ergab, dass die Studierenden die Facharbeit als die nützlichste Vorbereitung auf das Studium bewerten. Bislang wurde die Facharbeit in einem der beiden Leistungskursfächer angefertigt und ging mit maximal 30 Punkten in die Ab-

iturnote (insgesamt 840 Punkte) ein. Um die SchülerInnen besser auf das Studium vorzubereiten, wurde im Zuge der Reform der Oberstufe an bayerischen Gymnasien (G8) das W-Seminar geschaffen. Die Seminararbeit, die im Rahmen des W-Seminars angefertigt wird, löst die Facharbeit ab und fällt mit erreichbaren 60 von 900 Punkten stärker ins Gewicht als bisher. Dabei werden der Prozess der Erstellung, die schriftliche Arbeit und die Abschlußpräsentation bewertet. Die Abschlußpräsentation ersetzt die mündliche Prüfung über die Facharbeit. Das W-Seminar nimmt zwei Wochenstunden im Stundenplan der Ausbildungsabschnitte 11/1, 11/2 und 12/1 ein. Somit steht nun Unterrichtszeit für die Förderung wissenschaftlichen Arbeitens zur Verfügung. Im Vordergrund stehen dabei:

- fachwissenschaftliche Informationen
 - Recherche
 - Analyse und Abstraktionsfähigkeit
 - Reduktion auf wesentliche Inhalte
 - Strukturierung und Argumentation
 - korrektes und vielfältiges Präsentieren
- Finden kreativer Wege und Lösungen
- Einhaltung von Formalia und Zeitrahmen

Die SchülerInnen sollen mit Kompetenzen ausgestattet werden, die ihnen für Studium und Berufsleben von Nutzen sind. Das W-Seminar ist als Heranführung an die Arbeitsweise der Hochschulen gedacht. Die SchülerInnen sollen lernen, wie bei der Erstellung einer wissenschaftlichen Arbeit präzise und fachlich korrekt gearbeitet wird. Zu den Kompetenzen zählt auch der Einsatz und das eigene Erstellen von Medien. Fachliche Präzisierungen sollen durch einen Kompetenzkatalog der jeweiligen Fachschaft spezifiziert werden.

6.1.2 Leitfach, Lehrereinsatz und Rahmenthema

Jedes W-Seminar ist einem Leitfach zugeordnet und wird von einer Lehrkraft betreut. Es besteht die Möglichkeit bei fächerverbindenden oder fächerübergreifenden Themen, auch weitere Lehrkräfte zur Betreuung hinzuziehen. Das Rahmenthema muss grundlegend zwei Anforderungen erfüllen:

1. Es muss thematisch so weit gefasst sein, dass sich genügend Themen für Semi-

narbeiten ergeben.

2. Es muss thematisch so eng gefasst sein, dass die Bearbeitung der Themen Möglichkeiten zur Kooperation und Diskussion unter den SchülerInnen liefert. Obendrein müssen sich alle Ergebnisse in einen Gesamtzusammenhang einordnen lassen.

Das Rahmenthema kann entweder auf Themen vorhergehender Jahrgangsstufen aufbauen, Themen aus der 11. und 12. Jahrgangsstufe vertiefen oder vollkommen vom Fachlehrplan losgelöst sein. Die Bedingung ist jedoch, dass sich charakteristische Methoden des jeweiligen Unterrichtsfachs daran erlernen lassen. Eine zu enge Anlehnung des Rahmenthemas an den Lehrplan des Fachunterrichts ist zu vermeiden, da die SchülerInnen nicht verpflichtet sind, auch das Leitfach zu besuchen.

6.1.3 Methodik und Zeitplan

Den Kern des W-Seminars sollen Unterrichtsmethoden bilden, die eigenständiges Arbeiten bzw. forschendes Lernen fordern und fördern. Beispiele dafür sind:

- Quellenarbeit, Textarbeit und Werkanalyse
- empirisches Arbeiten
- Versuch und Experiment
- Werkerstellung
- Referat/Präsentation und Diskussion
- Exkursionen und Erkundungen
- Expertenreferate
- individuelles Arbeiten und Arbeiten im Team

Um Exkursionen, Vorträge, etc. leichter durchführbar zu machen und den Fachunterricht dadurch nicht zu beeinträchtigen, sind für das W-Seminar Randstunden vorgesehen. Zudem soll die Unterrichtszeit flexibel gehandhabt werden, damit Veranstaltungen des Seminars auch ausserhalb der Schulzeit durchgeführt werden können. Auf die zeitliche Gesamtbelastung der SchülerInnen muss jedoch stets geachtet werden.

Zeitlicher Ablauf des W-Seminars:

Ausbildungsabschnitt 10/2

Die SchülerInnen wählen ein W-Seminar aus dem Angebot der Schule.

Ausbildungsabschnitt 11/1

In der „Input-Phase“ erfolgt die Einführung in das Rahmenthema und das wissenschaftliche Arbeiten. Neben dem Herausarbeiten geeigneter Seminararbeitsthemen stehen auch erste eigene Recherchen im Vordergrund.

Ausbildungsabschnitt 11/2

Die SchülerInnen arbeiten nun eigenständig an ihren Themen und legen zu bestimmten Terminen Zwischenergebnisse vor, welche auch in der Gruppe diskutiert werden. Dabei ist auf eine gute Beratung und Begleitung durch die Lehrkraft zu achten.

Ausbildungsabschnitt 12/1

Die Abgabe der Seminararbeiten erfolgt nach den Herbstferien. Anschließend findet die Vorbereitung der Abschlußpräsentation und eine abschließende Einordnung der Ergebnisse in den Gesamtzusammenhang statt. Unterdessen wird die Arbeit am Rahmenthema ebenfalls fortgeführt.

Jeder Teilnehmer wählt sein individuelles Seminararbeitsthema zum Ende des Ausbildungsabschnitts 11/1, wobei der endgültige Titel erst während 11/2 festgelegt werden muss. Um die SchülerInnen bei ihrer eigenständigen Arbeit zu unterstützen, ist auf eine bedachte Begleitung und Beratung des Lern- und Arbeitsprozesses durch die Lehrkraft zu achten. Dabei ist es wichtig sowohl Einzelgespräche zu führen als auch Zwischenberichte im Seminar zu erörtern. Die Zwischenberichte dienen der Lehrkraft dazu, sich ein Bild über den Stand der Arbeiten zu verschaffen, und zwingen SchülerInnen mit mangelndem Zeitmanagement zur Einhaltung des Zeitplans.

6.1.4 Lernort Bibliothek

Eine bedeutende Funktion im W-Seminar kommt dem Lern- und Arbeitsort Bibliothek zu. Die SchülerInnen sollen lernen, sich Informationen aus verschiedenen Medien zu beschaffen und sie weiterzuverarbeiten.

Schulbibliothek

Eine zeitgemäß ausgestattete Schulbibliothek sollte jedem Seminaristen Zugriff auf Printmedien, audiovisuelle und digitalisierte Medien sowie auf das Internet erlauben. Sie bietet den SchülerInnen Möglichkeiten zu verschiedensten Arbeitsformen wie z.B. entdeckendem, handlungsorientiertem, fächerübergreifendem, sozialem und multimedialen Arbeiten. Außerdem bietet sie der Lehrkraft die Chance, die SchülerInnen intensiv und individuell zu betreuen. Da die Schulbibliothek auch außerhalb der Seminarstunden eine zentrale Rolle im W-Seminar spielen soll, sind angemessene Öffnungszeiten notwendig.

Öffentliche und wissenschaftliche Bibliotheken

Je nach Region besteht auch die Option, eine öffentliche oder wissenschaftliche Bibliothek als Lernort zu nutzen. Über die jeweilige Bibliotheks-Versorgung gibt es Informationen unter:

<http://www.informationskompetenz.de/regionen/bayern/kontakt-fuer-schulen/>.

Oftmals werden von den Bibliotheken spezielle Workshops für SchülerInnen zu den Themen Literaturrecherche und professionelle Internetrecherche angeboten.

6.1.5 Leistungserhebung

Die Leistungen der SchülerInnen aus den Ausbildungsabschnitten 11/1 und 11/2 gehen als Halbjahresleistungen (wie in anderen Fächern) in das Abiturzeugnis ein. Dabei sind mindestens zwei kleine Leistungsnachweise pro SchülerIn und Halbjahr zu erbringen. Anzahl, Form, Bewertungskriterien und Gewichtung der Nachweise werden mit Genehmigung der Schulleitung festgelegt und den SchülerInnen zu Beginn des Seminars erläutert. Die Seminararbeit und die Abschlußpräsentation fließen als eigenständige Bewertung in die Abiturnote ein. In 12/1 finden keine Leistungserhebungen statt, damit sich die SchülerInnen auf ihre Seminararbeit konzentrieren können. Da vor allem in 11/1 neue fachliche und methodische Grundkompetenzen erworben werden, sollte die Zahl der Leistungsüberprüfungen gering gehalten werden und die Lernorientierung des Unterrichts im Vordergrund stehen.

Anfangs sollten vertraute Formen der Leistungserhebung (z.B. Stegreifaufgaben, Abfragen,...) bevorzugt werden. Es dürfen jedoch auch Leistungen bewertet werden, die

im Rahmen von Arbeitsaufträgen, also auch in häuslicher Arbeit, erbracht werden. Grundsätzlich gilt, dass nur geprüft werden kann, was mit den SchülerInnen vorbereitet wurde.

In 12/1 findet die Bewertung des schriftlichen Teils der Seminararbeit und der Abschlußpräsentation statt. Sie wird im Verhältnis 3:1 gewichtet. Die Arbeit wird unter den Gesichtspunkten Inhalt, Darstellung und Formalia nach dem Schema 3:2:1 benotet. Bei der Bewertung der Darstellung ist auf eine sinnvolle Gliederung, Übersichtlichkeit und sprachlich-grammatikalische Korrektheit zu achten. Unter Formalia sind Aspekte zu verstehen, die sich direkt auf die schriftliche Arbeit beziehen (z.B. richtige Zitierweise, äußere Form). Bei der Bewertung der Abschlußpräsentation sollte eine Gewichtung von Darbietung und Inhalt/Aufbau im Verhältnis 1:1 gewählt werden.

6.1.6 Konzepterstellung durch die Lehrkraft

Bevor eine Schule ein W-Seminar anbietet, muss durch eine Lehrkraft ein Konzept erstellt werden. Aus diesem gehen

- Zeitplan
- Tätigkeiten der Lehrkraft und der SchülerInnen in den verschiedenen Arbeitsphasen
- die geplanten Leistungserhebungen und
- die dafür vorgesehenen Bewertungskriterien

hervor. Neben dem Rahmenthema müssen mindestens sechs Beispiele für Seminararbeitsthemen angegeben werden. Ob das Konzept den SchülerInnen zur Wahl gestellt wird, entscheidet die Schulleitung. Die Teilnahme an Wettbewerben ist grundsätzlich möglich. Dabei können Wettbewerbsbeiträge Seminararbeiten ersetzen (Anerkennung erfolgt durch das Kultusministerium).

6.1.7 Wahlprozess zur Einrichtung der Seminarkurse

Der Auswahlprozess ist in 3 Phasen gegliedert und wird von den OberstufenkoordinatorInnen organisiert. Phase 1 ist die Erstellung eines W-Seminar-Angebots durch die Schulen (siehe Abbildung 6.1). In Phase 2 wird jedes Seminar in einer Kurzbeschrei-

Jahrgangsstufe 10 – Oktober bis Februar				
Nr.	Prozess-Schritt	Ausführende(r)	benötigt	Ergebnis
1	Aufforderung an die Lehrkräfte, Seminare anzubieten	Oberstufenkoordinator/in, Fachbetreuer/in, Schulleiter/in	Merkblatt mit wesentlichen Informationen zum W-Seminar	Interesse der Lehrkräfte wecken
2	Erarbeitung von Seminarkonzepten durch interessierte Lehrkräfte	Lehrkräfte	Formblatt für Seminarkonzepte	Angebot verschiedener Seminare
3	Überprüfen der Seminarkonzepte	Fachbetreuer/in, Oberstufenkoordinator/in, Schulleiter/in	Seminarkonzepte mit den erforderlichen Angaben	Bewertung der Seminarangebote
4	Entscheidung über das Angebot von Seminaren	Schulleiter/in	Bewertung der Seminarangebote	Angebot der Schule an W-Seminaren

Abbildung 6.1: Einrichtung der W-Seminare - Phase 1 [MR08]

bung den SchülerInnen vorgestellt (siehe Abbildung 6.2). Die Wahl der W-Seminare (Erst- und Zweitwunsch) durch die SchülerInnen findet im Februar statt. Die Semina-

Jahrgangsstufe 10 – Februar				
Nr.	Prozess-Schritt	Ausführende(r)	benötigt	Ergebnis
5	Bekanntgabe des Seminarangebots	Oberstufenkoordinator/in	Aushang der Kurzbeschreibungen	Information der Schüler/innen
6	Beratung der Schüler/innen	Oberstufenkoordinator/in, Fachlehrkräfte, Schüler/innen	Gespräche mit den Schüler/innen	Entscheidungshilfen für die Schüler/innen
7	Wahl eines Seminars durch die Schüler/innen (Erst- und Zweitwunsch)	Oberstufenkoordinator/in, Schüler/innen	Wahlzettel	Anzahl der potenziellen Seminarteilnehmer/innen
8	Auswahl der Seminare	Oberstufenkoordinator/in, Schulleiter/in	Wahlergebnisse, Budgetinformationen	Seminare der Schule

Abbildung 6.2: Einrichtung der W-Seminare - Phase 2 [MR08]

re sollten eine Teilnehmerzahl von 15 nicht überschreiten. Die Schulleitung beschließt anschließend bis zur Abgabe der „vorläufigen Unterrichtsübersicht“ (meist im April), welche Seminare eingerichtet werden. Die Berücksichtigung des Schulbudgets und die Vereinbarkeit des Seminars mit dem Schulprofil fließen ebenfalls in die Entscheidung ein. Durch die Entscheidung der Schulleitung, welche SchülerInnen letztendlich welches Seminar besuchen dürfen, wird die Phase 3 abgeschlossen. Dabei wird auf eine maximale Übereinstimmung mit den Schülerwünschen geachtet.

6.1.8 Maßnahmen zur Qualitätssicherung

Damit qualitativ hochwertiger Unterricht stattfindet, müssen von der Schulleitung geeignete Lehrkräfte ausgewählt werden und das Seminarkonzept gut durchdacht sein. Die Leistungserhebungen sind ein wichtiger Anhaltspunkt für die Unterrichtsqualität, dabei muss im Vorfeld auf Transparenz (Bewertungskriterien) geachtet werden. Da es für das W-Seminar keinen vorgegebenen Lehrplan gibt, liegt die Verantwortung für die Qualitätssicherung bei den Fachschaften und der Schulleitung. Durch die ständige Betreuung der SchülerInnen kann auf die Qualität des Seminars auch kurzfristig Einfluss genommen werden. Als Richtlinien stehen die im Konzept und Zeitplan verankerten Meilensteine zur Verfügung, die der Lehrkraft Aufschluss über die Diskrepanz zwischen den Zielen des Seminars und dem derzeitigen Stand geben. Wichtigste Quelle zur Verbesserung des Unterrichts ist das Feedback der SchülerInnen zum Ablauf des Seminars. Zudem spiegeln auch die Güte der Seminararbeiten und Abschlußpräsentationen den Stand des Seminars wider.

6.2 Umsetzung

Im Folgenden werden Vorschläge zur praktischen Umsetzung der Vorgaben aus Abschnitt 6.1 gegeben. Es handelt sich dabei lediglich um einen Leitfaden zu den einzelnen Punkten, der jederzeit erweitert und verbessert werden kann.

6.2.1 Ziel: Kompetenz in wissenschaftlichem Arbeiten

Das W-Seminar „Kosmische Strahlung“ gibt den Seminaristen die Chance, sich mit einem hochaktuellen Thema auseinanderzusetzen und neueste fachwissenschaftliche Informationen zu recherchieren. Das Gebiet der kosmischen Strahlung ist nicht im bayerischen Gymnasial-Lehrplan enthalten und ist daher für die SchülerInnen völliges „Neuland“. Durch die Experimente mit dem CRMD, deren Auswertung und durch die Einordnung der Ergebnisse wird den SchülerInnen ein tiefer Einblick in den Physikeralltag gegeben. Während des Studiums wird das Beherrschung von Computerprogrammen für wissenschaftliches Arbeiten meist von den Dozenten vorausgesetzt. Die Erstellung und Präsentation der Seminararbeit erfordert eine genaue Auseinandersetzung mit verschiedenen Programmen zur Tabellenkalkulation, Erstellung von Diagrammen, Textsatz, etc. und gibt den SchülerInnen eine solide methodische Basis für das Studium.

6.2.2 Leitfach, Lehrereinsatz und Rahmenthema

Das Leitfach zum W-Seminar „Kosmische Strahlung“ ist natürlich Physik. Die Auswahl der möglichen Seminararbeiten ist breit gefächert. Zum Einen bietet der CRMD eine Zahl von Experimenten, welche als Seminararbeiten zur Auswahl stehen, zum Anderen bleibt die Chance, eine rein theoretische Arbeit über bestimmte Aspekte der kosmischen Strahlung zu verfassen. Vor allem die experimentellen Seminararbeiten erfordern gute Kooperation unter den SchülerInnen. Da pro Schule im Regelfall nur ein Detektorset vorhanden sein wird, ist schon allein wegen des zeitlichen Aspekts Absprache nötig.

6.2.3 Methodik und Zeitplan

Das W-Seminar „Kosmische Strahlung“ bietet vielfältige Möglichkeiten für eigenständiges Arbeiten und forschendes Lernen. Bei der Erstellung der Seminararbeit ist individu-

elles Arbeiten gefragt. Da sich das Thema nicht im Lehrplan befindet, ist es in keinem Schulbuch enthalten und fordert somit eigene Quellenarbeit von den SchülerInnen. Da große Teile der Literatur in englischer Sprache verfasst sind, müssen die SchülerInnen fächerübergreifend lernen. Dies ist eine gute Vorbereitung auf das Studium/Berufsleben, wo die Beherrschung des Englischen vorausgesetzt wird. Das Experimentieren mit dem CRMD-Set bildet das Herzstück des Seminars. Es ermöglicht den SchülerInnen eigene Versuche zu einem Gegenstand aktueller Forschung und fordert von ihnen Teamfähigkeit und Organisation. Da ständig Zwischenberichte über die Seminararbeiten abgelegt werden müssen, wird auch das Erstellen von Referaten und Präsentationen geübt. Aufgrund der Komplexität einiger Teilaspekte der kosmischen Strahlung ist es unmöglich, dass sich alle SchülerInnen detailliert damit befassen. Vielmehr sollten diese Themen als Referate (siehe Abschnitt 6.2.5) vergeben werden. Die Bewertung dieser Präsentationen fließt in die Benotung des Seminars ein.

Damit die SchülerInnen ein realistisches Bild vom Berufsbild „PhysikerInnen“ und einem „professionellen“ Experiment erhalten, ist beispielsweise eine Exkursion zum „KASKADE-GRANDE“-Experiment in Karlsruhe denkbar. Auch bieten sich zu speziellen Teilgebieten des Themas Expertenreferate von Dozenten der Universität Erlangen-Nürnberg an. Die Nähe zu dieser erleichtert die Organisation solcher Vorträge.

Ein Vorschlag zum zeitlichen Ablauf des W-Seminars wird in Abschnitt 6.3 erläutert.

6.2.4 Lernort Bibliothek

Inzwischen sind alle Gymnasien mit ausreichend vielen Internet-Arbeitsplätzen und Druckmöglichkeiten ausgestattet. Dies ermöglicht es den SchülerInnen im Netz nach Informationen zu recherchieren, selbst wenn sie zu Hause über keinen Internetanschluss verfügen. Das Internet ist die wohl ergiebigste Quelle, da inzwischen sämtliche wissenschaftliche Publikationen online zum Download bereitstehen. Dabei ist jedoch auf ein den SchülerInnen angemessenes Niveau der Arbeiten zu achten. Eine Liste mit Publikations-Archiven wird in Abschnitt 6.4.2 vorgestellt.

Es ist sehr unwahrscheinlich, dass eine Schulbibliothek geeignete Literatur zum Thema „kosmische Strahlung“ bereithält. Deshalb wäre es begrüßenswert, wenn die entsprechenden Bibliotheken um folgende Werke (siehe auch Abschnitt 6.4.1) erweitert werden könnten:

- Claus Grupen - Astroparticle physics
- Helmut Zimmermann - ABC Astronomie
- Alfred Weigert - Astronomie und Astrophysik

Sollte eine Beschaffung nicht aus Schulmitteln finanziert werden können, besteht auch die Chance auf Bezuschussung durch eine physikalische Stiftung (Haereus, etc.), sowie durch lokale Stiftungen (Siemens-Stiftung, Fördervereine der Schulen, etc.).

Alternativ kann auch das Angebot der Universitätsbibliothek Erlangen-Nürnberg genutzt werden. SchülerInnen ab der 11. Klasse ist es möglich, einen Benutzerausweis an einer der Ausleihstellen durch Vorlage eines Personalausweises oder Reisepasses zu beantragen. Die Termine der Einführungen in die Bibliotheksnutzung können den Aushängen in der Hauptbibliothek (Schuhstraße 1a) entnommen werden statt. Die Literatur zum Seminarthema befindet sich zu größten Teilen in der Teilbibliothek GP09 (Physik). Viele Titel können auch über das Online-System „Opacplus“ (<http://www.opac.uni-erlangen.de>) bestellt werden.

6.2.5 Leistungserhebung

Grundsätzlich sind sämtliche Formen der Leistungserhebung zulässig. Die Einführung in dieses neue, weitgehend unbekannte Themengebiet stellt für die SchülerInnen eine Herausforderung dar, welche anfangs nicht noch zusätzlich durch neue Arten der Leistungserhebung erschwert werden sollte. Allerdings sollte im Verlauf des Seminars zur Notengebung durch universitätsübliche Prüfungsformen übergegangen werden. Damit sind weniger mehrstündige Klausuren oder mündliche Prüfungen gemeint, sondern vielmehr eigenständig erarbeitete Vorträge, Präsentation und Schriftwerke. Diese im Studium ständig auftretenden Formen der Leistungserhebung stellen für viele Studenten zu Beginn die größten Hürden dar und sollten daher bereits in der Schule trainiert werden. Damit geht auch eine Auseinandersetzung mit Programmen zur Erstellung von digitalen Präsentationen und Gestaltung von Grafiken und Texten einher. Eine Beherrschung dieser Techniken wird an den Universitäten meist vorausgesetzt, beziehungsweise das Nicht-Beherrschen stets als Mangel der schulischen Ausbildung kritisiert.

Lehrkraft: _____		Leitfach: _____	
Rahmenthema: _____			
Zielsetzung des Seminars, Begründung des Themas (ggf. Bezug zum Fachprofil): _____			
Halb- jahre	Monate	Tätigkeit der Schülerinnen/Schüler und der Lehrkraft	geplante Formen der Leistungserhebung und Bewertungskriterien
11/1	Sept. – Dez.		
	Jan. – Feb.		
11/2	März – April		
	Mai – Juli		
12/1	Sept. – Nov.		Seminararbeit
	Dez. – Feb.		Präsentation
mögliche Themen für die Seminararbeiten (bitte mindestens sechs Themen angeben): 1. _____ 2. _____ 3. _____ 4. _____ 5. _____ 6. _____			
weitere Bemerkungen zum geplanten Verlauf des Seminars: _____			

Abbildung 6.3: Formular zur Beantragung eines W-Seminars [MR08]

6.2.6 Konzepterstellung durch die Lehrkraft

Abbildung 6.2.6 zeigt das Formular zur Beantragung eines W-Seminars. Das Rahmenthema des W-Seminars ist die „kosmische Strahlung“, das selbstverständlich dem Leitfach Physik zugeordnet ist. Ziel des Seminars ist es, einen detaillierten Einblick in die Arbeitsalltag von PhysikerInnen zu erlangen und sich mit dem Forschungsgebiet der kosmischen Strahlung, welches noch viele unbeantwortete Fragen beherbergt, auseinanderzusetzen. Dabei sollen die SchülerInnen auch Einblick in die Struktur des Universums bekommen.

Die drei Halbjahre 11/1, 11/2 und 12/1 werden jeweils in zwei Phasen aufgeteilt. Bei der Beantragung eines W-Seminars müssen für jede dieser Phasen die Tätigkeiten von Schü-

lerInnen und Lehrer und die Formen der Leistungserhebungen (siehe Abschnitt 6.2.5) angegeben werden. Das Experimentieren mit dem CRMD ist ein fester Bestandteil des Unterrichts. Ein Leitfaden für das thematische Vorgehen im Unterricht findet sich in Abschnitt 6.3.

Mögliche Themen für praktische Seminararbeiten sind beispielsweise:

- Kalibration und zeitliche Veränderung der Detektorcharakteristiken (TOT, Zählrate, ...)
- Auswirkung von Luftdruckschwankungen auf die Myonen-Zählrate des CRMD
- Auswirkung des geomagnetischen Ost-West-Effekts
- Messung der Myonen-Lebensdauer und Methoden zur Reduktion der Untergrundrauschens
- Messung der Zenitwinkelabhängigkeit von Myonen
- Untersuchung der Effizienz verschiedener Detektorgeometrien bei der Luftschauer-Messung
- Vermessung des Mondschantens anhand von CRMD-Datensätze
- Bau einer schwenkbaren Halterung für das CRMD-Set
- Untersuchung von Koinzidenzen innerhalb des Schularrays

Aber auch SchülerInnen, welchen eine theoretische Arbeit mehr zusagt, bieten sich Themen wie zum Beispiel:

- Funktionsweise und Aufbau des „Pierre-Auger-Observatoriums“
- Das „Global-Positioning-System“
- Theorie der Fermi-Beschleunigungsprozesses
- Ist die Herkunft kosmischer Strahlung isotrop oder anisotrop? - Bewertung neuester Erkenntnisse
- Typen und Physik „aktiver galaktischer Kerne“ (AGN)
- Funktionsweise von Sekundärelektronenvervielfachern

Weitere Themen sind auf der Homepage des WALTA-Projekts zu finden: <http://neutrino.phys.washington.edu/~walta/webresearchtopics.html>.

Das W-Seminar „Kosmische Strahlung“ lebt von einer engen Kooperation mit dem „Erlangen Centre for Astroparticle Physics“ (**ECAP**). Dieser direkte Kontakt zu den

SchülerInnen muss genutzt werden, um sie für das Studienfach Physik zu begeistern. Als vorläufige Ansprechpartner fungieren hier Prof. Christian Stegmann und Dr. Kathrin Valerius.

6.2.7 Wahlprozess zur Einrichtung der Seminarkurse

In den Wahlprozess der SchülerInnen kann nicht aktiv eingegriffen werden. Es gilt jedoch im Vorfeld den LehrerInnen gute Vorschläge für Konzepte zum W-Seminar-Thema „kosmische Strahlung“ zu liefern (siehe Beispiel in Abschnitt 6.2.6) und vor allem ihr Interesse an der Thematik zu wecken. Dabei muss den LehrerInnen immer genug eigener Spielraum zur Gestaltung des Seminars gelassen werden. Wird versucht die LehrerInnen um jeden Preis in ein vorgefertigtes Konzept zu zwingen, kann dies zu Ablehnung führen. Vielmehr sollte gemeinsam auf eine aktive und kooperative Gestaltung geachtet werden, um durch didaktisches Geschick (Lehrerseite) und Fachwissen (Universität) die Stärken beider Seiten zu vereinen.

Hat sich die Schulleitung entschlossen, das W-Seminar „Kosmische Strahlung“ anzubieten, gilt es nun die SchülerInnen dafür zu begeistern. Zusätzlich zur Kurzbeschreibung des Seminars bieten sich vor allem Poster und Broschüren als Werbung für dieses an. Diese sollten vom ECAP gestaltet werden. Dabei sollten altersgerecht (die Zielgruppe ist 16/17 Jahre alt) einige interessante Aspekte und Fragen vorgestellt werden. Vielleicht weckt auch ein Vortrag von ECAP-Mitgliedern zu diesem Thema (eventuell mit Vorstellung des CRMD) die nötige Neugier der Schüler.

6.2.8 Maßnahmen zur Qualitätssicherung

Vor allem bei der ersten Durchführung des Seminars sollte auf permanentes Feedback zwischen Schulen und Universität geachtet werden. Hier muss zudem eine durchgehende technische Betreuung der Schulen durch das ECAP gewährleistet werden, um die LehrerInnen mit dem „unvertrauten“ Aufbau nicht im Stich zu lassen. Schon während des Kurses sollten Probleme bei der Kommunikation zwischen den Partnern erkannt und beseitigt werden. Am Ende des Seminars sollten nochmals die Schwächen in der Zusammenarbeit von Schulen und Universität analysiert werden. Wahrscheinlich ist ein persönliches Gespräch mit den betreuenden Lehrkräften die sinnvollste Form der Rückmeldung.

6.3 Vorschlag zum Aufbau eines Lehrplans

In diesem Abschnitt wird eine Möglichkeit zum Aufbau des W-Seminars „Kosmische Strahlung“ vorgestellt. Dabei wird sich dem Thema auf historischem Weg genähert, indem zuerst die Entdeckungen von Viktor Hess und Pierre Auger behandelt werden. Anschließend wird genauer auf Art und Physik der ankommenden Teilchen eingegangen. Dies geschieht durch eine Einführung in das Standardmodell der Teilchenphysik. Methoden zur Messung der Teilchen sollen im darauffolgenden Unterrichtsblock erläutert werden. Nach diesen Vorarbeiten wird nun auf den Kern des Themas eingegangen: die Entstehung von Sekundär- und Primärteilchen der kosmischen Strahlung.

Das Vorwissen der SchülerInnen sollte folgende Punkte umfassen:

1. Energie und ihre Formen
2. Spezielle Relativitätstheorie
3. Bewegung elektrisch geladener Teilchen in elektro-magnetischen Feldern
4. Teilchen, Kerne und Zerfallsprozesse

Entdeckung der kosmischen Strahlung

In den ersten Unterrichtseinheiten wird die Entdeckung der kosmischen Strahlung durch Viktor Hess und Pierre Auger skizziert. Dabei werden die Fragen und Erwartungen der beiden Forscher dargelegt. Auch wenn die ganze Physik erst im Block über den Nachweis von Teilchen verstanden werden kann, wird trotzdem auf den Aufbau der Experimente eingegangen. Die Ergebnisse ihrer Messungen bilden gleichzeitig die Einleitung des Seminars.

Standardmodell der Teilchenphysik

Die nächsten Unterrichtseinheiten sollten sich mit der Physik der in der kosmischen Strahlung gemessenen Teilchen befassen. Dabei wird nochmals beim Atomaufbau begonnen und anschließend in immer kleinere Dimensionen übergegangen. Besonderes Augenmerk liegt dabei auf den Teilchen, welche in der kosmischen Strahlung am häufigsten auftreten: π^\pm , μ^\pm , e^\pm , γ , $\nu/\bar{\nu}$. Da Pionen und Myonen nicht stabil sind, werden hier ebenfalls Zerfallsprozesse behandelt.

Nachweis von Teilchen

Es gibt verschiedenste Methoden, um die nun kennengelernten Teilchen nachzuweisen. Dazu wird zunächst auf die Wechselwirkungen von Teilchen und die spezielle Relativitätstheorie eingegangen. Als Detektionsmöglichkeiten werden der Szintillationsprozess, die Fluoreszenzmethode und die Cherenkov-Strahlung vorgestellt. In diesem Zusammenhang werden Sekundärelektronenvervielfacher und Koinzidenzschaltungen, wie sie auch im CRMD-Set verwendet werden, behandelt. In diesem Teil ist auf die Vor- und Nachteile der verschiedenen Methoden und die Kombination mehrerer einzugehen.

Sekundärstrahlung

Das nächste Teilgebiet des Themas ist die Sekundärstrahlung. Sie wird in der Atmosphäre durch Auftreffen der Primärstrahlung auf Atomkerne initiiert und dehnt sich in ihrem Verlauf durch weitere Wechselwirkungen und Zerfälle zu einem Luftschauer aus. Dabei ist sie aus der hadronischen, myonischen und elektro-magnetischen Komponente aufgebaut. Diese kann auf der Erde, wie von Hess und Auger erstmals gezeigt wurde, gemessen werden (zum Beispiel mit dem CRMD-Set).

Primärstrahlung

Das abschließende Teilgebiet grenzt nun direkt an Interessen aktueller Forschung. Primärteilchen sind Atomkerne mit hohen Energien. Hier wird zuerst auf das Energiespektrum der Primärteilchen und ihre Elementhäufigkeit eingegangen. Woher aber stammen diese Teilchen und was hat sie so stark beschleunigt? Im weiteren Verlauf gilt es mögliche Beschleunigungsmechanismen aufzuarbeiten und ihren Gültigkeitsbereich zu ermitteln. Auch Kandidaten für die Quellen der Teilchen und die Einflüsse bei der Propagation sind zu beleuchten.

6.4 Literatur für Schüler zum Thema kosmische Strahlung

Hier soll ein Überblick über die Literatur zur kosmischen Strahlung gegeben werden. Viele Werke sind auf Englisch verfasst. Daher stellen vor allem die Fachbegriffe SchülerInnen und LehrerInnen vor Probleme. Zum Nachschlagen sollte hier auf Online-Angebote zurückgegriffen werden. Geeignete Anlaufstellen sind beispielsweise die Online-Lexika <http://dict.leo.org> oder <http://dict.cc>.

6.4.1 Printmedien

Die kosmische Strahlung wird in einer ganzen Reihe von Büchern thematisiert. Es ist nicht möglich hier sämtliche aufzuführen, hier sollen jedoch einige, für Schüler geeignete Exemplare erwähnt werden. Hierbei muss vor allem das Buch „Astroparticle Physics“ von Claus Grupen hervorgehoben werden. Die englische Neuauflage hat sich gegenüber der älteren deutschen Auflage stark verbessert. Obwohl das Buch in englischer Sprache verfasst ist, bietet es einen systematischen Einblick in die kosmische Strahlung. Zur Sekundärstrahlung lässt sich in Thomas K. Gaissers „Cosmic rays and particle physics“ viel Material finden. Ein praktisches Nachschlagewerk zu astronomischen Begriffen ist das „ABC Astronomie“ von Helmut Zimmermann. Zwei Standardwerke zur Astronomie dürfen hier ebenfalls nicht fehlen: „Astronomie und Astrophysik“ von Alfred Weigert und „Fundamental Astronomy“ von Hannu Karttunen. Weitere Hintergründe zu kosmischen Objekten sind unter anderem in Peter Schneiders „Einführung in die extragalaktische Astronomie und Kosmologie“ nachzulesen. Auch im „Astroparticle Physics“-Magazin sind oft Publikationen zu Aspekten der kosmischen Strahlung zu finden.

6.4.2 Online-Medien

Das Internet ist als Informationsquelle zum Thema kosmische Strahlung unerschöpflich. Grundsätzlich können Auskünfte zu den meisten Themen mit den üblichen Suchmaschinen gefunden werden. Nachfolgend sind einige Beispiele aufgeführt, die jedoch nur eine kleine Auswahl des Angebots darstellen.

Als Einführung in neue Themen ist die Seite <http://www.weltderphysik.de> für die

SchülerInnen stets eine gute Anlaufstelle. Sie wird von der Deutschen Physikalischen Gesellschaft und dem Bundesministerium für Bildung und Forschung herausgegeben und zielt damit speziell auf interessierte SchülerInnen ab. Ein sehr schöner Einstieg in die Astroteilchenphysik ist auch die vom „Komitee für Astroteilchenphysik“ herausgegebene Broschüre „Kosmische Spurensuche“. Diese steht unter <http://www.astroteilchenphysik.de> zum Download bereit. Eine geeignete Einleitung in die Teilchenphysik ist unter <http://www.teilchenphysik.de> zu finden. Von der Verlagsgesellschaft des Spektrums der Wissenschaft werden speziell für Mittel- und OberstufenschülerInnen aufbereitete Materialien online unter <http://www.wissenschaft-schulen.de> bereitgestellt.

Viele Informationen zur kosmischen Strahlung sind in Online-Publikationsarchiven verfügbar. Als äußerst ergiebig stellte sich die Seite <http://arxiv.org> heraus. Aber auch unter <http://en.scientificcommons.org> und <http://www.aanda.org> gibt es Materialien zum Thema.

Auf den Internetseiten vieler Experimente sind ebenfalls Informationen (manchmal sogar Datensätze) erhältlich. Sehr übersichtlich und gut strukturiert ist die deutsche Seite der „Pierre-Auger-Kollaboration“ (<http://www.auger.de>). Das „Kaskade-Grande-Experiment“ in Karlsruhe hält auf seiner Homepage (siehe Literaturliste in Abschnitt 6.4.3) ebenso viele Information über kosmische Strahlung bereit.

6.4.3 Literaturliste

Abschließend werden hier nochmals alle Bücher und Internetseiten aufgeführt, die im Rahmen dieser Arbeit gesichtet und für schülertauglich befunden wurden. Diese Liste ist keinesfalls vollständig und sollte ständig überarbeitet und erweitert werden.

Printmedien

Titel	Autor	Verlag	Jahr	Kosten
Astroparticle Physics	Claus Grupen	Springer	2005	53,45 €
Cosmic Bullets: High Energy Particles In Astro-physics	Roger Clay	Basic Books	1998	16,00 \$
Cosmic rays and particle physics	Thomas K. Gaisser	Cambridge University Press	1991	45,99 €
ABC Astronomie	Helmut Zimmermann	Spektrum Akademischer Verlag	2008	34,95 €
Astronomie und Astrophysik: Ein Grundkurs	Alfred Weigert	Wiley-VCH Verlag	2009	59,00 €
Fundamental Astronomy	Hannu Karttunen	Springer	2007	52,46 \$
Einführung in die extragalaktische Astronomie und Kosmologie	Peter Schneider	Springer	2007	59,95 €

Internetseiten

Titel	Adresse	Herausgeber
Welt der Physik	http://www.weltderphysik.de	DPG, BMBF
Astroteilchenphysik in Deutschland	http://www.astroteilchenphysik.de	Jürgen Eschke et al.
Teilchenphysik in Deutschland	http://www.teilchenphysik.de	Deutsches Elektronen-Synchrotron
wissenschaft in die schulen!	http://www.wissenschaft-schulen.de	Spektrum der Wissenschaft Verlagsgesellschaft
arXiv.org e-Print archive	http://arxiv.org	Cornell University Library
Scientific Commons	http://en.scientificcommons.org	magazine.One UG
Astronomy&Astrophysics	http://www.aanda.org	EDP Sciences
Pierre-Auger-Observatorium	http://www.auger.de	Karlsruher Institut für Technologie
KASCADE-Grande	http://www-ik.fzk.de/KASCADE	Karlsruher Institut für Technologie

7 Organisation

Die Durchführung des W-Seminars zur kosmischen Strahlung an den Erlanger Schulen geschieht in Kooperation mit dem ECAP (Ansprechpartner: Prof. Christian Stegmann), was für dieses auch einige Aufgaben mit sich bringt. Da die Thematik auch für die LehrerInnen Neuland darstellt, müssen diese auf dem Gebiet der Astroteilchenphysik fortgebildet werden. Auch der Zusammenbau der Detektorsets sollte durch Mitarbeiter des Instituts begleitet werden. Die Koordination der Zeiträume zur Datennahme mit den CRMD-Sets als ausgedehntem Array und schnelle Hilfe bei technischen Problemen sollte ebenfalls durch das ECAP erfolgen. Außerdem stellen gelegentliche Treffen mit den Schulen eine gute Möglichkeit dar, um direkt auf Fragen/Probleme einzugehen und weiteres Interesse bei den SchülerInnen zu erzeugen.

7.1 Lehrerfortbildung - Astroteilchenphysik

Als zusätzlichen Anreiz zur Auseinandersetzung mit dem Thema kann vom ECAP eine Lehrerfortbildung organisiert werden. Diese hat die Vorteile für die Lehrkräfte, dass die Anreise (im Gegensatz zu manchen anderen Maßnahmen) keinerlei Probleme bereitet und der Einstieg in das Thema kosmische Strahlung erleichtert wird. Für das ECAP bietet sich der Vorteil der direkten Begegnung mit den LehrerInnen und somit der Möglichkeit der gezielten Beantwortung auftretender Fragen. Die Entscheidung über die Anrechnung einer besuchten Veranstaltung auf die individuelle Fortbildungsverpflichtung von Lehrkräften wird von der Schulleitung getroffen. Dabei ist die reine Dauer der Veranstaltung kein grundsätzliches Kriterium für die Anerkennungsfähigkeit. Passt es zur Fortbildungsplanung der Schule, so können beispielsweise einstündige, halbtägige, eintägige oder mehrtägige Veranstaltungen anerkannt werden [FIB].

7.2 Betreuung Detektorbau

Detektoren wie das CRMD-Set gehören normalerweise nicht zum Repertoire der „Physik-Sammlung“ an Schulen. Damit steigt das Risiko, irreparable Schäden an den teuren Komponenten zu verursachen. Um dies zu vermeiden, sollte den Detektorbau anfangs ein Institutsmitarbeiter unterstützend begleiten. Die Grundlage für ein zuverlässiges Experimentieren mit dem CRMD ist ein vernünftig zusammengebauter Detektor. Mit der Erfahrung der ECAP-Mitglieder sollte dieses Unterfangen kein Problem darstellen. Außerdem können somit erste Fragen zum Aufbau des Systems (z.B. Photomultiplier, Szintillator, ...) direkt von einem Experten beantwortet werden.

7.3 Betreuung Detektorarray

Die Synchronisation der Datennahme des Detektorarrays sollte vom ECAP koordiniert werden. Dazu ist genaue Rücksprache mit den Betreuungslehrkräften der beteiligten Schulen zu halten, um geeignete Zeiträume zu finden (mögliche Probleme durch Ferien, Klassenfahrten, ...). Bei technischen Schwierigkeiten muss schnell und unkompliziert geholfen werden können. Dazu ist die Einrichtung eines ständigen Ansprechpartners seitens des ECAPs sinnvoll. Dieser muss für die Schulen stets erreichbar sein. Außerdem bietet sich die Gelegenheit, einmal pro Semester ein Meeting der W-Seminar-Teilnehmer aller Schulen an der Universität zu veranstalten. Diese Möglichkeit des direkten Kontakts zu den SchülerInnen kann genutzt werden, um auf fachliche Fragen einzugehen, Expertenvorträge zu hören und Ergebnisse einzelner Schüler-Gruppen vorzustellen. Auch können die SchülerInnen so untereinander Erfahrungen austauschen und ihre Resultate vergleichen. Die Einrichtung eines Online-Portals für dieses Projekt würde Koordination und Feedback zwischen den Partnern sehr erleichtern.

8 Fazit

Die vorliegende Arbeit gibt einen weiten Überblick über das CRMD-Set, seine Möglichkeiten und einen Vorschlag zur Gestaltung eines W-Seminars zum Thema „kosmische Strahlung“. Der Aufbau und der Einstieg in die Handhabung des Sets werden durch die Erklärungen (Kapitel 2) in dieser Arbeit erleichtert. Hier stiftet leider das QuarkNet Handbuch [RJPB10] einige Verwirrung, da sich die Zeitauflösung der neuen Board-Generation geändert hat und dies leider in der aktuellen Version (Januar 2010) nicht an allen Stellen entsprechend korrigiert wurde.

Obwohl sehr viel Messzeit durch Fehlersuche eingebüßt wurde, ist dies dennoch positiv zu bewerten: die Ursachen vieler Probleme konnten identifiziert werden und auf mögliche Beseitigungsmöglichkeiten wird in dieser Arbeit an den betroffenen Stellen jeweils hingewiesen. Die Datennahme lief zum Ende der Messungen hin relativ störungsfrei ab. Um dies weiter zu verbessern, sollte das Terminalprogramm `getdata` (Abschnitt 4.2.1) dahingehend modifiziert werden, dass es Störungen selbst erkennt und behandelt.

Die hier durchgeführten Experimente sind dem QuarkNet Handbuch [RJPB10] entnommen und stellen eine wichtige Säule dieser Arbeit dar. Damit ist die Fülle der Möglichkeiten noch lange nicht ausgeschöpft. Anregungen für weitere Versuche kommen in Abschnitt 3.5 zur Sprache.

Viel Mühe kostete die Entwicklung der Java-Werkzeuge (Kapitel 4). Diese Programme erfüllten sämtliche Aufgaben im Rahmen dieser Arbeit zuverlässig. Somit eignen sie sich als Grundgerüst für künftige Verbesserungen und Modifikationen.

Der Vorschlag zur Gestaltung des W-Seminars zum Thema der kosmischen Strahlung dient als erster Leitfaden für die damit betrauten Lehrkräfte. Es wurde dabei bewusst darauf geachtet, diesen nicht zu festgelegt zu formulieren. Die LehrerInnen müssen ihren Unterricht individuell gestalten können und dürfen nicht zur Umsetzung eines „zementierten“ Konzepts gezwungen werden. Dies könnte sich nämlich negativ auf die Motivation der Lehrkräfte auswirken. Sie müssen weiterhin die Rolle des Gestalters

innehaben. Hierbei ist mit „Gestalter“ jedoch nicht gemeint, dass als Unterrichtsform ausschließlich Frontalunterricht in Frage kommt. Vielmehr ist auf eine angepasste und ausgewogene Wahl der Unterrichtsformen mit viel Raum für Schüleraktivität zu achten. Dazu bieten sich die Experimente und Arbeiten mit dem CRMD-Set hervorragend an.

Dem ECAP wird bei der Umsetzung des W-Seminar-Konzepts eine besonders wichtige Rolle zuteil. Die Thematik übersteigt das während der Lehrerausbildung Erlernte bei Weitem. Somit ist die Universität gefordert, den Lehrkräften die Sachverhalte verständlich näherzubringen. Nur wenn die LehrerInnen mit den Themen vertraut sind, können sie diese auch überzeugend an die SchülerInnen weitervermitteln. Dazu kommt die durchgehende technische Hilfestellung für die Schulen. Auftretende Probleme müssen unverzüglich und möglichst unkompliziert behoben werden, ohne dass dadurch eine zusätzliche Belastung für die Lehrkräfte entsteht. Gleichzeitig bietet das Projekt die Chance, die Werbetrommel für die Physik als Studienfach zu rühren. Werbung ist auch während der Phase sinnvoll, in der die SchülerInnen ihre W-Seminare wählen. Hierzu bietet sich wohl am besten ein „Cosmic-Ray-Day“ mit Präsentationen und Versuchen zu diesen Themen für die Zehntklässler der Erlanger Schulen an.

A Board-Befehle

HE

Quarknet Scintillator Card, Qnet2.5 Vers 1.11 Compiled Jul 15 2009 HE=Help
Serial#=6459 uC_Volts=3.34 GPS_TempC=22.8 mBar=1018.9

CE - TMC Counter Enable.
CD - TMC Counter Disable.
DC - Display Control Registers, (C0-C3).
WC a d - Write Control Registers, addr(0-6) data byte(H).
DT - Display TMC Reg, 0-3, (1=PipeLineDelayRd, 2=PipeLineDelayWr).
WT a d - Write TMC Reg, addr(1,2) data byte(H), if a=4 write delay word.
DG - Display GPS Info, Date, Time, Position and Status.
DS - Display Scalar, channel(S0-S3), trigger(S4), time(S5).
RE - Reset complete board to power up defaults.
RB - Reset only the TMC and Counters.
SB p d - Set Baud,password, 1=19K, 2=38K, 3=57K ,4=115K, 5=230K, 6=460K, 7=920K
SA n - Save setup, 0=(TMC disable), 1=(TMC enable), 2=(Restore Defaults).
TH - Thermometer data display (@ GPS), -40 to 99 degrees C.
TL c d - Threshold Level, signal ch(0-3)(4=setAll), data(0-4095mV), TL=read.
Veto - Veto select, Off='VE 0', On='VE 1', Gate='VG c', 0-255(D) 10ns/cnt.
View - View setup registers. Setup=V1, Voltages(V2), GPS LOCK(V3).
HELP - HE,H1=Page1, H2=Page2, HB=Barometer, HS=Status, HT=Trigger.

H2

Barometer Qnet Help Page 2

BA - Display Barometer trim setting in mVolts and pressure as mBar.
BA d - Calibrate Barometer by adj. trim DAC ch in mVlts (0-4095mV).

Flash

FL p - Load Flash with Altera binary file(*.rbf), p=password.
FR - Read FPGA setup flash, display sumcheck.
FMR p - Read page 0-3FF(h), (264 bytes/page)
Page 100h= start fpga *.rbf file, page 0=saved setup.

GPS

NA 0 - Append NMEA GPS data Off,(include 1pps data).
NA 1 - Append NMEA GPS data On, (Adds GPS to output).

A Board-Befehle

NA 2 - Append NMEA GPS data Off,(no 1pps data).
NM 0 - NMEA GPS display, Off, (default), GPS port speed 38400, locked.
NM 1 - NMEA GPS display (RMC + GGA + GSV) data.
NM 2 - NMEA GPS display (ALL) data, use with GPS display applications.

Test Pulser

TE m - Enable run mode, 0=Off, 1=One cycle, 2=Continuous.
TD m - Load sample trigger data list, 0=Reset, 1=Singles, 2=Majority.
TV m - Voltage level at pulse DAC, 0-4095mV, TV=read.

Serial #

SN p n - Store serial # to flash, p=password, n=(0-65535 BCD).
SN - Display serial number (BCD).

Status

ST - Send status line now. This resets the minute timer.
ST 0 - Status line, disabled.
ST 1 m - Send status line every (m) minutes.(m=1-30, def=5).
ST 2 m - Include scalar data line, chs S0-S4 after each status line.
ST 3 m - Include scalar data line, plus reset counters on each timeout.
TI n - Timer (day hr:min:sec.msec), TI=display time, (TI n=0 clear).
U1 n - Display Uart error counter, (U1 n=0 to zero counters).
VM 1 - View mode, 0x80=Event_Demarcation_Bit outputs a blank line.
- View mode returns to normal after 'CD','CE','ST' or 'RE'.

HB

BA - Display barometer data as raw counts(BCD) and mBar.
BA bbbb - Calibrate by setting trim DAC Ch voltage(0-4095mV).

HS

Trigger IRQ Status Byte, bit assignments (see "HT" for location on data line)

0x1 = Unused
0x2 = Trigger FIFO full (1Hz uC STAT LED flash increases if FIFO_full)
0x4 = Unused
0x8 = Current or last 1PPS tick not within 25MHz +/-50 nSec

Status Line Format for BCD1-BCD19, see "ST" command.

BCD1 -> mBar
BCD2 -> GPS_DegC (format nn.n)
BCD3 -> 1PPS Delay (mSec, hardware pulse to ASCII NMEA data)
BCD4 -> CPU_Vcc(3.3v) (mVolts)
BCD5 -> GPS_UTC
BCD6 -> GPS_DATE
BCD7 -> GPS_VALID (A=Valid)
BCD8 -> GPS_SAT Count
BCD9 -> 1PPS Time

A Board-Befehle

```
BCD10->   Code Version   (format n.nn)
BCD11->   Serial #
BCD12-15> TMCregs  3-0   (4 tmc regs displayed as a 32 Bit#)
BCD16-19> Cntlregs 3-0   (4 cntrl regs displayed as a 32 Bit#)
```

```
ST 1008 +273 +086 3349 180022 021007 A 05 C5ED5FF1 106 6148 00171300 000A710F
  mBar   | 1ppsDly | GpsUTC   | GpsVld | 1ppsTime | SerNum   |      Cntlregs
        GpsDegC   CPU_Vcc   GpsDate   GpsSat#   CodeVer   TMCregs
```

```
DS 000006B4 00001413 00000D62 000006B1 00001414
  Ch0Cnts Ch1Cnts Ch2Cnts Ch3Cnts TrigCnt
```

HT

```
Timer Counter Bits 31..0   ( 8 bytes ascii)
RE0 TAG  RE0 DATA          ( 2 bytes ascii) --
FE0 TAG  FE0 DATA          ( 2 bytes ascii) |
RE1 TAG  RE1 DATA          ( 2 bytes ascii) | "0x80=Event_Demarcation_Bit"
FE1 TAG  FE1 DATA          ( 2 bytes ascii) | "0x20=Edge_Tag_Bit"
RE2 TAG  RE2 DATA          ( 2 bytes ascii) | "0x1F=Data,(5 bits)"
FE2 TAG  FE2 DATA          ( 2 bytes ascii) |
RE3 TAG  RE3 DATA          ( 2 bytes ascii) |
FE3 TAG  FE3 DATA          ( 2 bytes ascii) --
1PPS TIME Bits 31..0       ( 8 bytes ascii)
GPS RMC UTC  hhmmss.sss    (10 bytes ascii)      Status Flag bits
GPS RMC DATE ddmmyy        ( 6 bytes ascii)      0x1 = Unused
GPS RMC STATUS A=valid     ( 1 byte  ascii)      0x2 = Trigger FIFO full
GPS GGA SATELLITES USED    ( 2 bytes ascii)      0x4 = Unused
TRIG IRQ STATUS FLAGS     ( 1 byte  ascii)      0x8 = 1PPS > +/-50nSec
GPS 1PPS to DATA DELAY mS ( 5 bytes ascii)
```

GPS Status

```
Example data line with GPS receiving 5 satellites. |
DE799F14 BB 00 00 00 00 00 00 00 DE1C993A 132532.010 111007 A 05 0 +0060
DE799F15 00 00 00 00 21 00 00 00 DE1C993A 132532.010 111007 A 05 0 +0060
DE799F15 00 35 00 00 00 00 00 00 DE1C993A 132532.010 111007 A 05 0 +0060
DE799F15 00 00 00 00 00 3C 00 00 DE1C993A 132532.010 111007 A 05 0 +0060
|_____| | ch0 | ch1 | ch2 | ch3 | |_____| |_____| |_____| | | | _1ppsDelay
Timer RE-FE RE-FE RE-FE RE-FE 1pps Hex GPS Time Date | | _Status Flag
40nS/cnt 1.25nS/cnt(5 bits) DMY | _Satellites Used
```

View Mode 1 example

```
ED608136 3100 0 3700 0 3976416186 132542.010 111007 A 05 0 +0060
ED608137 2C 0 0 0 3976416186 132542.010 111007 A 05 0 +0060
ED608137 0 0 32 0 3976416186 132542.010 111007 A 05 0 +0060
Timer ch0 ch1 ch2 ch3 1pps Dec
```

B Java-Quelltexte

B.1 autocal

```
1  /**
2   * autocal
3   */
4
5  import java.io.*;
6  import gnu.io.*;
7  import java.awt.*;
8  import java.awt.event.*;
9
10 public class autocal
11 {
12     //gnu.io: COM-Port initialisieren
13     static CommPortIdentifier portID;
14     private static final long serialVersionUID = 1L;
15     static SerialPort serps;
16     static String port,output;
17
18     //In- und Outputstreams
19     InputStream ins;
20     static OutputStream out;
21     FileOutputStream fos;
22
23     //Wartezeit bei der Kalibration
24     static float wait;
25
26     public static void main(String[] args)
27     {
28         if (args.length != 3) {
29             System.out.println("java_ autocal_<COM#>_<interval_ [m]>_<outputfile>");
30             System.exit(1);
31         }
32         port = args[0];
33         try {
34             wait = Float.parseFloat(args[1]);
35         } catch (NumberFormatException exc) {
36             System.out.println("Keine_Fliesskommazahl:_ " + args[1] + "\n" + exc);
37         }
38         output = args[2];
39
40         //COM-Port holen
41         try {
42             portID = CommPortIdentifier.getPortIdentifier(port);
```

B Java-Quelltexte

```
43     serss = (SerialPort) portID.open("autocal",10000);
44     autocal f = new autocal();
45 } catch (PortInUseException exc) {
46     System.out.println("COM-Port_" + port + "_ist_besetzt.\n" + exc);
47 }
48     catch (NoSuchPortException exc) {
49         System.out.println("Kein_COM-Port_mit_Kennung:_" + port + "\n" + exc);
50     }
51     catch (NullPointerException exc) {
52         System.out.println("Fehler:_" + exc);
53     }
54 }
55
56 public autocal()
57 {
58     //Streams initialisieren
59     try {
60         ins = serss.getInputStream();
61         out = serss.getOutputStream();
62         serss.addEventListener(new commListener());
63     } catch (Exception e) {
64         System.out.println("Fehler:_" + e);
65     }
66     serss.notifyOnDataAvailable(true);
67
68     //Parameter für Schnittstelle: Baud-Rate, 8 Bits, 1 Stopbit, keine Parität
69     try {
70         serss.setSerialPortParams(115200, SerialPort.DATABITS_8,
71                                     SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
72     } catch (UnsupportedCommOperationException e) {}
73
74     //Threshold-Level von 600 - 0mV durchlaufen
75     for (int i = 600; i >= 0; i = i - 25) {
76         PrintWriter aus = new PrintWriter(out, true);
77         aus.append("TL_4_" + i + "\r");
78         aus.append("RB" + "\r");
79         aus.flush();
80
81         //Thread aussetzen
82         try {
83             Thread.sleep((int) (wait * 60000f));
84         } catch (InterruptedException e) {}
85         aus.append("DS" + "\r");
86         aus.close();
87     }
88
89     //TL auf 600mV, um PMT zu schonen
90     PrintWriter aus = new PrintWriter(out, true);
91     aus.append("TL_4_600\r");
92     aus.flush();
93     aus.close();
94     System.exit(1);
95 }
96
97 //Klasse liest COM-Port-Stream in Datei output ein
98 public class commListener implements SerialPortEventListener
99 {
100     public void serialEvent(SerialPortEvent event)
101     {
```

```

102     try {
103         fos = new FileOutputStream(output,true);
104         if (event.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
105             byte[] readBuffer = new byte[1000];
106             while (ins.available() > 0) {
107                 int numBytes;
108                 while ((numBytes = ins.read(readBuffer)) > 0) {
109                     fos.write(readBuffer,0,numBytes);
110                 }
111             }
112         }
113         fos.close();
114         ins.close();
115     } catch (IOException e) {
116         System.out.println("Fehler: " + e);
117     }
118 }
119 }
120 }

```

B.2 getdata

```

1  /*
2   *getdata
3   */
4
5  import java.io.*;
6  import gnu.io.*;
7  import java.awt.*;
8  import java.awt.event.*;
9
10 public class getdata extends Frame
11 {
12     //gnu.io: COM-Port initialisieren
13     static CommPortIdentifier portID;
14     private static final long serialVersionUID = 1L;
15     static SerialPort serps;
16     static String port,output;
17
18     //In- und Outputstreams
19     InputStream ins;
20     OutputStream out;
21     FileOutputStream fos;
22
23     //Fensterelemente
24     TextArea ausgabe;
25     TextField tfe;
26     Button bt;
27
28     public static void main(String[] args)
29     {
30         if (args.length != 2) {
31             System.out.println("java -getdata <COM#> <outputfile>");
32             System.exit(1);
33         }
34         port = args[0];
35         output = args[1];

```

B Java-Quelltexte

```
36
37 //COM-Port öffnen
38 try {
39     portID = CommPortIdentifier.getPortIdentifier(port);
40     serss = (SerialPort) portID.open("getdata",10000);
41     getdata f = new getdata();
42     f.setSize(330,250);
43     f.setVisible(true);
44 } catch (PortInUseException exc) {
45     System.out.println("COM-Port_" + port + "_ist_besetzt.\n" + exc);
46 }
47     catch (NoSuchPortException exc) {
48     System.out.println("Kein_COM-Port_mit_Kennung:_ " + port + "\n" + exc);
49 }
50     catch (NullPointerException exc) {
51     System.out.println("Fehler:_ " + exc);
52 }
53 }
54
55 public getdata() {
56     //Fenster initialisieren
57     this.addWindowListener(new WindowAdapter() {
58         public void windowClosing(WindowEvent e)
59         {
60             dispose();
61             System.exit(0);
62         }
63     });
64     setLayout(new FlowLayout());
65     add(tfe=new TextField(40));
66     tfe.addActionListener(new sendeListener());
67     add(bt=new Button("Senden"));
68     bt.addActionListener(new sendeListener());
69     add(ausgabe=new TextArea("Port_" + port + "_geöffnet\n",7,40));
70     ausgabe.setEditable(false);
71
72     //Streams initialisieren
73     try {
74         ins = serss.getInputStream();
75         out = serss.getOutputStream();
76         serss.addEventListener(new commListener());
77     } catch (Exception e) {
78         System.out.println("Fehler:_ " + e);
79     }
80     serss.notifyOnDataAvailable(true);
81
82     //Parameter für Schnittstelle: Baud-Rate, 8 Bits, 1 Stopbit, keine Parität
83     try {
84         serss.setSerialPortParams(115200,SerialPort.DATABITS_8,
85                                     SerialPort.STOPBITS_1,SerialPort.PARITY_NONE);
86     } catch (UnsupportedCommOperationException e) {}
87 }
88
89 //Klasse sendet Eingabebefehle an COM-Port
90 public class sendeListener implements ActionListener
91 {
92     public void actionPerformed(ActionEvent e)
93     {
94         PrintWriter aus = new PrintWriter(out,true);
```



```

95     try {
96         fos = new FileOutputStream(output,true);
97         fos.close();
98     } catch (Exception ex) {
99         System.out.println(ex);
100    }
101    ausgabe.append(tfe.getText().toUpperCase() + "\r\n");
102    aus.append(tfe.getText() + "\r");
103    aus.close();
104    tfe.setText("");
105 }
106 }
107
108 //Klasse liest COM-Port-Stream in Datei output ein
109 public class commListener implements SerialPortEventListener
110 {
111     public void serialEvent(SerialPortEvent event)
112     {
113         try {
114             fos = new FileOutputStream(output,true);
115             if (event.getEventType() == SerialPortEvent.DATA_AVAILABLE) {
116                 byte[] readBuffer = new byte[1000];
117                 while (ins.available() > 0) {
118                     int numBytes;
119                     while ((numBytes = ins.read(readBuffer)) > 0) {
120                         fos.write(readBuffer, 0, numBytes);
121                     }
122                 }
123             }
124             fos.close();
125             ins.close();
126         } catch (IOException e) {
127             System.out.println("Fehler: "+e);
128         }
129     }
130 }
131 }

```

B.3 linecheck

```

1  /*
2   *linecheck
3   */
4
5  import java.io.*;
6
7  public class linecheck
8  {
9      public static void main(String[] args)
10     {
11         String line,input,output = "dummy";
12         int u,o,c = 0;
13
14         //Lesen, Schneiden, zu Datum springen
15         boolean read = false, cut = false, jump = false;
16
17         if (args.length == 3) {

```

```

18     read = true;
19 }
20 if (args.length == 4) {
21     if (args[3].equals("t") == true) {
22         jump = true;
23     }
24     else {
25         cut = true;
26         output = args[3];
27     }
28 }
29 if (args.length != 4 && args.length != 3) {
30     System.out.println("Lesen: \java_linecheck_<inputfile>_<startline>" +
31         "\_<endline>");
32     System.out.println("Schneiden: \java_linecheck_<inputfile>_<startline>" +
33         "\_<endline>_<outputfile>");
34     System.out.println("Springen_zu_Zeit/Datum: \java_linecheck_<inputfile>" +
35         "\_<hhmmss>_<ddmmyy>_t");
36     System.exit(1);
37 }
38 input = args[0];
39 u = Integer.valueOf(args[1]).intValue();
40 o = Integer.valueOf(args[2]).intValue();
41 try {
42     LineNumberReader f = new LineNumberReader(new FileReader(input));
43
44     //Lesemodus
45     if (read == true) {
46         while ((line = f.readLine()) != null) {
47             c++;
48             if (f.getLineNumber() >= u && f.getLineNumber() <= o) {
49                 System.out.println(f.getLineNumber() + ":\_ + line);
50             }
51         }
52         System.out.println("Gesamtzahl_der_Zeilen:\_ + c);
53     }
54
55     //Schneidemodus
56     if (cut == true) {
57         FileWriter ou = new FileWriter(output);
58         while ((line = f.readLine()) != null) {
59             if (u <= f.getLineNumber() && f.getLineNumber() <= o) {
60                 ou.write(line + "\r\n");
61             }
62             if (f.getLineNumber() > o) {
63                 break;
64             }
65         }
66         ou.close();
67     }
68
69     //Datumsmodus
70     if (jump == true) {
71         while ((line = f.readLine()) != null) {
72             if (line.length() > 59 && line.substring(42, 48).compareTo(args[1]) == 0
73                 && line.substring(53, 59).compareTo(args[2]) == 0) {
74                 c++;
75                 System.out.println(f.getLineNumber() + ":\_ + line);
76             }

```

```

77     }
78     System.out.println("Gesamtzahl der Zeilen mit " + args[1] + " " + args[2]
79                         + ":" + c);
80 }
81 f.close();
82 } catch (IOException e) {
83     System.out.println("Fehler beim Lesen der Datei!");
84 }
85 }
86 }

```

B.4 fill

```

1  /*
2   * fill
3   */
4
5  import java.io.*;
6
7  public class fill
8  {
9      public static void main(String[] args)
10     {
11         String input,output;
12         String line,repl;
13         float c = 0f, io_c = 0f, filled_c = 0f;
14         boolean nio = false;
15
16         if (args.length != 2) {
17             System.out.println("java fill <inputfile> <outputfile>");
18             System.exit(1);
19         }
20         else {
21             input = args[0];
22             output = args[1];
23             try {
24                 //Streams initialisieren
25                 LineNumberReader f = new LineNumberReader(new FileReader(input));
26                 FileWriter ou = new FileWriter(output);
27                 FileWriter recycling = new FileWriter("recycling_" + output);
28                 while ((line = f.readLine()) != null) {
29                     c++;
30
31                     //Zeile okay UND keine Statuszeile ST
32                     if (line.length() == 72 && line.substring(0,6).indexOf("ST") == -1) {
33
34                         //Leerzeichen unter den ersten sechs Zeichen
35                         if (line.substring(0,6).indexOf("_") != -1) {
36                             nio = true;
37                         }
38
39                         //kein Leerzeichen unter den ersten sechs Zeichen
40                         else {
41
42                             //Repariere Zeitinformation durch Zeitinformation aktueller Zeile
43                             if (nio == true) {
44                                 repl = line.substring(0,6);

```

```

45         f.reset();
46         line = f.readLine();
47         line = repl + line.substring(6,72);
48         ou.write(line + "\r\n");
49         nio = false;
50         f.mark(500);
51         filled_c++;
52     }
53     else {
54         ou.write(line + "\r\n");
55         f.mark(500);
56     }
57 }
58 io_c++;
59 }
60
61 //Zeile okay UND Statuszeile ST
62 if (line.length() == 72 && line.substring(0,6).indexOf("ST") != -1) {
63     ou.write(line + "\r\n");
64     f.mark(500);
65     io_c++;
66 }
67
68 //Zeilenlänge muss 72 Zeichen sein, sonst in Recycling-Datei
69 if (line.length() != 72) {
70     recycling.write(line + "\r\n");
71 }
72 }
73 System.out.println("Anzahl_verworfenen_Zeilen:" + (c - io_c));
74 System.out.println("Anzahl_rekonstruierter_Zeilen:" + filled_c);
75 System.out.println("Anzahl_gelesener_Zeilen:" + c);
76 System.out.println("Anteil_verlorener_Daten:" + ((c - io_c) * 100 f
77     / c));
78 System.out.println("Anteil_rekonstruierter_Daten:" + (filled_c * 100 f
79     / c));
80 ou.close();
81 f.close();
82 recycling.close();
83 } catch (IOException e) {
84     System.out.println("Fehler_beim_Lesen_der_Datei!");
85 }
86 }
87 }
88 }

```

B.5 correct

```

1  /*
2   *correct
3   */
4
5  import java.io.*;
6
7  public class correct
8  {
9      public static void main(String[] args)
10     {

```

B Java-Quelltexte

```
11 String input,output,date_act;
12
13 if (args.length != 3) {
14     System.out.println("Erfordert_mit_'fill'_bearbeitete_Datei!");
15     System.out.println("java_correct_<inputfile><startdate:_ddmmyy><outputfile>");
16     System.exit(1);
17 }
18 else {
19     input = args[0];
20     date_act = args[1];
21     output = args[2];
22     try {
23
24         //Initialisiere Streams
25         LineNumberReader f = new LineNumberReader(new FileReader(input));
26         FileWriter ou = new FileWriter(output);
27         String line;
28
29         //Lese Input-Datei ein
30         while ((line = f.readLine()) != null) {
31
32             //Datum aktualisieren bei Tageswechsel
33             if (line.substring(42,48).equals("000000") == true) {
34                 date_act = line.substring(53,59);
35             }
36
37             //falsches Datum ersetzen
38             if (line.substring(42,48).equals("235959") == true &&
39                 later(line.substring(53,59),date_act) == true &&
40                 line.length()>59) {
41                 line = line.replaceAll(line.substring(53,59),date_act);
42             }
43
44             //Schreibe in Output-Datei
45             ou.write(line + "\r\n");
46         }
47         ou.close();
48         f.close();
49     } catch (IOException e) {
50         System.out.println("Fehler_beim_Lesen_der_Datei!");
51     }
52 }
53 }
54
55 //Methode zum Vergleich von Datumsinformationen: TRUE, falls rawdate neuer
56 public static boolean later(String rawdate, String actual) {
57     int rday,aday,rmonth,amonth,ryear,ayear;
58     rday = Integer.parseInt(rawdate.substring(0,2));
59     aday = Integer.parseInt(actual.substring(0,2));
60     rmonth = Integer.parseInt(rawdate.substring(2,4));
61     amonth = Integer.parseInt(actual.substring(2,4));
62     ryear = Integer.parseInt(rawdate.substring(4,6));
63     ayear = Integer.parseInt(actual.substring(4,6));
64     if (rday > aday && rmonth >= amonth && ryear >= ayear) {
65         return true;
66     }
67     if (rmonth > amonth && ryear >= ayear) {
68         return true;
69     }
}
```

```

70     if (ryear > ayear) {
71         return true;
72     }
73     else {
74         return false;
75     }
76 }
77 }

```

B.6 sort

```

1  import java.io.*;
2
3  public class sort
4  {
5      public static void main(String[] args)
6      {
7          String line, input;
8          int count = 0;
9          if (args.length != 1) {
10             System.out.println("Benutzung: fill:~java~sort~<inputfile>~");
11             System.exit(1);
12         }
13         else {
14             input = args[0];
15             try {
16                 LineNumberReader f = new LineNumberReader(new FileReader(input));
17
18                 //Zeilenanzahl ermitteln
19                 while ((line = f.readLine()) != null) {
20                     count++;
21                 }
22                 f.close();
23                 System.out.println("Anzahl der Zeilen:~" + count);
24
25                 //Einlesen von Zeit und Zeilennummer der Daten
26                 long[] time = new long[count];
27                 int[] zeile = new int[count];
28                 String[] data = new String[count];
29                 for (int i = 0; i < count; i++) {
30                     zeile[i] = i + 1;
31                 }
32                 LineNumberReader g = new LineNumberReader(new FileReader(input));
33                 while ((line = g.readLine()) != null) {
34                     time[g.getLineNumber() - 1] = Long.parseLong
35                         (line.substring(0, line.indexOf("~")));
36                     data[g.getLineNumber() - 1] = line;
37                 }
38                 g.close();
39
40                 //chronologisches Sortieren der Daten
41                 sortiere(time, zeile);
42
43                 //Ausgabe der sortierten Daten in Datei
44                 FileWriter ou = new FileWriter("sorted" + args[0]);
45                 for (int i = 0; i < count; i++) {
46                     ou.write(data[zeile[i]-1] + "\r\n");

```

```

47     }
48     ou.close();
49 } catch (IOException e) {
50     System.out.println("Fehler beim Lesen der Datei");
51 }
52 }
53 }
54
55 //Datensortierung mit dem BubbleSort-Verfahren
56 public static void sortiere(long[] t,int[] x) {
57     boolean unsortiert = true;
58     long temp;
59     int tmp;
60     while (unsortiert) {
61         unsortiert = false;
62         for (int i = 0; i < t.length-1; i++) {
63             if (t[i] > t[i+1]) {
64                 temp = t[i];
65                 tmp = x[i];
66                 t[i] = t[i+1];
67                 x[i] = x[i+1];
68                 t[i+1] = temp;
69                 x[i+1] = tmp;
70                 unsortiert = true;
71             }
72         }
73     }
74 }
75 }

```

B.7 readout

```

1  /*
2   *readout
3   */
4
5  import java.io.*;
6
7  public class readout
8  {
9      public static void main(String[] args)
10     {
11         String line,input,output;
12         long[] start = new long[4];
13         long[] tot = new long[4];
14         if (args.length != 2) {
15             System.out.println("java_readout_<inputfile>_<outputfile>");
16             System.exit(1);
17         }
18
19         //Kommandozeile einlesen
20         input = args[0];
21         output = args[1];
22         try {
23
24             //Streams initialisieren
25             LineNumberReader f = new LineNumberReader(new FileReader(input));

```

B Java-Quelltexte

```

26     FileWriter ou = new FileWriter(output);
27     FileWriter[] Split = new FileWriter[4];
28     for (int i = 0; i <= 3; i++) {
29         Split[i] = new FileWriter(args[1].substring(0, args[1].indexOf(".")) +
30             "CH" + i + ".txt");
31     }
32
33     //Daten einlesen
34     while ((line = f.readLine()) != null) {
35         if (real(line) != -1) {
36             for (int i = 0; i <= 3; i++) {
37                 if (lese(line, real(line), i, 0) != null) {
38                     start[i] = Long.parseLong(lese(line, real(line), i, 0));
39                 }
40                 if (lese(line, real(line), i, 3) != null && start[i] != 0) {
41                     tot[i] = (Long.parseLong(lese(line, real(line), i, 3)) - start[i]);
42                     ou.write("" + start[i] + "_" + tot[i] + "_" +
43                         line.substring(53, 59) + "_" + i + "\r\n");
44                     Split[i].write("" + start[i] + "_" + tot[i] + "_" +
45                         line.substring(53, 59) + "_" + i + "\r\n");
46                 }
47             }
48         }
49         else {}
50     }
51     ou.close();
52     for (int i = 0; i <= 3; i++) {
53         Split[i].close();
54     }
55     f.close();
56 } catch (IOException e) {
57     System.out.println("Fehler beim Lesen der Datei!");
58 }
59 }
60
61 //Echtzeit in ps
62 public static long real(String line)
63 {
64     try {
65         long result = (Long.parseLong(line.substring(0, 8), 16) -
66             Long.parseLong(line.substring(33, 41), 16)) * 400001 +
67             10000000000001 * (Long) (Math.round(3600d *
68             Double.parseDouble(line.substring(42, 44)) + 60d *
69             Double.parseDouble(line.substring(44, 46)) +
70             Double.parseDouble(line.substring(46, 52)) +
71             Double.parseDouble(line.substring(67, 72)) / 1000d));
72         return result;
73     } catch (NumberFormatException e) {
74         return -11;
75     }
76 }
77
78 //Echtzeit der Rising/Falling Edges, RE: rf = 0, FE: rf = 3
79 public static String lese(String line, long echt, int i, int rf)
80 {
81     String result = null;
82     String RE = line.substring(9 + rf + i * 6, 11 + rf + i * 6);
83     if (RE.substring(0, 1).equals("2") || RE.substring(0, 1).equals("A")) {
84         long c = echt + (Long.parseLong(RE.substring(1, 2), 16)) * 12501;

```


B Java-Quelltexte

```
85     result = "" + c;
86 }
87 if (RE.substring(0,1).equals("3") || RE.substring(0,1).equals("B")) {
88     long c = echt + (16l + Long.parseLong(RE.substring(1,2),16)) * 12501;
89     result = "" + c;
90 }
91 else {}
92 return result;
93 }
94
95 //Triggerkanal, NICHT FERTIG
96 public static int trigger(String line, long[] start)
97 {
98     int triggerch = -1;
99     long current = 8640100000000000001;
100    if (line.substring(9,10).equals("8") || line.substring(9,10).equals("9") ||
101        line.substring(9,10).equals("A") || line.substring(9,10).equals("B")) {
102        for (int i = 0; i <= 3; i++) {
103            if (start[i] <= current) {
104                current = start[i];
105                triggerch = i;
106            }
107            else {}
108        }
109    }
110    else {}
111    return triggerch;
112 }
113 }
```

B.8 gnutot

```
1  /*
2   *gnutot
3   */
4
5  import java.io.*;
6
7  public class gnutot
8  {
9      public static void main(String[] args)
10     {
11         double bin, limit;
12         int number, c = 0;
13         String line, input, output;
14         if (args.length != 4) {
15             System.out.println("Benutzung: java gnutot <inputfile> +
16                                     <binwidth>[ns]> <limit>[ns]> <outputfile>");
17             System.exit(1);
18         }
19
20         //Kommandozeile einlesen
21         input = args[0];
22         bin = Double.parseDouble(args[1]);
23         limit = Double.parseDouble(args[2]);
24         output = args[3];
25         number = (int) (Math.round(limit / bin)) + 1;
```

```

26  int[] bins = new int[number];
27  try {
28
29      //Streams initialisieren
30      LineNumberReader f = new LineNumberReader(new FileReader(input));
31      FileWriter ou = new FileWriter(output);
32
33      //Daten einlesen
34      while ((line = f.readLine()) != null) {
35          c++;
36          if (line.indexOf("E") != -1) {
37
38              //TOT einlesen
39              try {
40                  double data = Double.parseDouble(line.substring(line.indexOf("_") + 1,
41                                                                line.indexOf("E") + 3));
42
43                  //TOT Intervall zuordnen
44                  for (int i = 0; i < number; ) {
45                      if ((double) (i) * bin * 1E-9 <= data &&
46                          data < (double) (i + 1) * bin * 1E-9) {
47                          bins[i]++;
48                          break;
49                      }
50                      else {}
51                      i++;
52                  }
53              } catch (Exception ex) {
54                  System.out.println("Fehler:_ " + ex + "_Zeile:_ " + c);
55              }
56          }
57          else {}
58      }
59
60      //Intervalle und Treffer in Ausgabedatei schreiben
61      for (int i = 0; i < number; ) {
62          System.out.println((double) (i) * bin + "_bis_" +
63                             (double) (i + 1) * bin + "_ns:_ " + bins[i]);
64          ou.write((double) (i) * bin + "bis" + (double)(i + 1) * bin + "ns:_ " +
65                  bins[i] + "\r\n");
66          i++;
67      }
68      ou.close();
69      f.close();
70  } catch (IOException e) {
71      System.out.println("Fehler_beim_Lesen_der_Datei!");
72  }
73  }
74  }

```

B.9 rate

```

1  import java.io.*;
2
3  public class rate
4  {
5      public static void main(String[] args)

```

```

6  {
7  String line,input,output;
8  long bin,starttime = 0l,acttime,c = 0l,counts = 0l;
9  long scale = 600000000000000l;
10 double area;
11 boolean first = true;
12
13 if (args.length != 4) {
14     System.out.println("Benutzung_gnutot:_java_rate_<inputfile>_ " +
15         "<binwidth_[s]>_<detectorarea_[cm^2]>_<outputfile>");
16     System.exit(1);
17 }
18 input = args[0];
19 input = args[3];
20 bin = (long) (Double.parseDouble(args[1]) * 1E12d);
21 area = Double.parseDouble(args[2]) / 1E4d;
22 try {
23
24     //Streams initialisieren
25     LineNumberReader f = new LineNumberReader(new FileReader(args[0]));
26     FileWriter ou = new FileWriter(args[3]);
27
28     //Daten einlesen
29     while ((line = f.readLine()) != null) {
30         acttime = Long.parseLong(line.substring(0,line.indexOf("_")));
31
32         //Festlegen der Startzeit beim ersten Durchlauf
33         if (first == true) {
34             starttime = Long.parseLong(line.substring(0,line.indexOf("_")));
35             first = false;
36         }
37
38         //Treffer im Intervall summieren
39         if (starttime + c * bin <= acttime && acttime < starttime +
40             (c + 1l) * bin) {
41             counts++;
42         }
43
44         //Zeit und Treffer in Ausgabedatei schreiben
45         if (acttime > starttime + (c + 1l) * bin) {
46             ou.write("" + (starttime + c * bin) + "_" +
47                 (long) ((double) (counts * scale) /
48                     (double) (bin) / area) + "\r\n");
49             c++;
50             counts = 1l;
51         }
52     }
53     ou.close();
54     f.close();
55 } catch (IOException e) {
56     System.out.println("Fehler_beim_Lesen_der_Datei");
57 }
58 }
59 }

```

B.10 angle

B Java-Quelltexte

```
1  /*
2   *angle
3   */
4
5  import java.io.*;
6
7  class angle
8  {
9
10     //geographische Breite und Laenge
11     static double gb;
12     static double gl;
13
14     //Zenit-, Azimut-, Deklinations- und Rektaszensionwinkel als Zeichenkette
15     public static String angles(double x[],double y[],double z[],long t[],
16                                double utc,String date,boolean report)
17     {
18         String r = new String();
19         String rep = new String();
20
21         //Datum
22         double D,M,Y;
23         D = Double.parseDouble(date.substring(0,2));
24         M = Double.parseDouble(date.substring(2,4));
25         Y = 2000 + Double.parseDouble(date.substring(4,6));
26
27         //Sinus und Cosinus aus Differenzvektoren und Zeitunterschieden
28         double s[] = new double[2];
29         double c[] = new double[2];
30         s[0] = 2.99792458E-1 * (((double) (t[1] - t[0])) / 1000 / length(x,y);
31         s[1] = 2.99792458E-1 * (((double) (t[2] - t[0])) / 1000 / length(x,z);
32         c[0] = Math.sqrt(1 - sq(s[0]));
33         c[1] = Math.sqrt(1 - sq(s[1]));
34         rep += "\nZeit_1_0:" + (((double) (t[1] - t[0])) / 1000) +
35              "\r\nZeit_2_0:" + (((double) (t[2] - t[0])) / 1000) +
36              "\r\nSinus:" + s[0] + "\t" + s[1] + "\r\n" + "Cosinus:" +
37              c[0] + "\t" + c[1] + "\r\n";
38
39         //Bei Sinus > 1 keine Berechnung
40         if (s[0] > 1 || s[1] > 1) {}
41         else {
42
43             //Berechnung der Drehachsen
44             double d1[] = new double[3];
45             double d2[] = new double[3];
46             double zaxis[] = new double[3];
47             zaxis[0] = 0;
48             zaxis[1] = 0;
49             zaxis[2] = 1;
50             d1 = norm(kreuz(diff(y,x), zaxis));
51             d2 = norm(kreuz(diff(z,x), zaxis));
52             rep += "Drehachsen:\t" + d1[0] + "\t" + d2[0] + "\r\n" + "\t\t" + d1[1] +
53                  "\t" + d2[1] + "\r\n" + "\t\t" + d1[2] + "\t" + d2[2] + "\r\n";
54
55             //Rotationsmatrizen
56             double m1[][] = new double[3][3];
57             double m2[][] = new double[3][3];
58             m1[0][0] = c[0] + sq(d1[0]) * (1 - c[0]);
59             m1[0][1] = d1[0] * d1[1] * (1 - c[0]);
```

```

60 m1[0][2] = d1[1] * s[0];
61 m1[1][0] = d1[0] * d1[1] * (1 - c[0]);
62 m1[1][1] = c[0] + sq(d1[1]) * (1 - c[0]);
63 m1[1][2] = -d1[0] * s[0];
64 m1[2][0] = -d1[1] * s[0];
65 m1[2][1] = d1[0] * s[0];
66 m1[2][2] = c[0];
67
68 m2[0][0] = c[1] + sq(d2[0]) * (1 - c[1]);
69 m2[0][1] = d2[0] * d2[1] * (1 - c[1]);
70 m2[0][2] = d2[1] * s[1];
71 m2[1][0] = d2[0] * d2[1] * (1 - c[1]);
72 m2[1][1] = c[1] + sq(d2[1]) * (1 - c[1]);
73 m2[1][2] = -d2[0] * s[1];
74 m2[2][0] = -d2[1] * s[1];
75 m2[2][1] = d2[0] * s[1];
76 m2[2][2] = c[1];
77
78 //Vektoren die Ebene aufspannen
79 double r1[] = new double[3];
80 double r2[] = new double[3];
81 r1 = mm(m1, diff(y,x));
82 r2 = mm(m2, diff(z,x));
83 rep += "Ebenenvektoren:␣" + r1[2] + "\t" + r2[2] + "\n";
84
85 //Normalenvektor der Ebene
86 double n[] = new double[3];
87 n = norm(kreuz(r1,r2));
88
89 //Normalenvektor mit positivem z
90 if (n[2] < 0) {
91     n[0] = n[0] * (-1);
92     n[1] = n[1] * (-1);
93     n[2] = n[2] * (-1);
94 }
95 r += "" + n[2] + "\t";
96 rep += "Normalenvektor:␣" + n[0] + "␣" + n[1] + "␣" + n[2] + "\r\n";
97
98 //Horizontwinkel h, Richtungswinkel ph
99 double h, ph1, ph2, ph, az;
100 h = Math.toDegrees(Math.asin(n[2]));
101 rep += "Hoehe:␣" + h + "\n";
102 ph = -Math.PI / 2;
103 if (n[0] == 0 && n[1] > 0) {
104     ph = Math.PI / 2;
105 }
106 if (n[0] == 0 && n[1] <= 0) {
107     ph = -Math.PI / 2;
108 }
109 if (n[0] != 0) {
110
111     //1. Quadrant ph = ph
112     if (n[0] > 0 && n[1] >= 0) {
113         ph = Math.atan(n[1] / n[0]);
114     }
115
116     //2. Quadrant ph = 180 + ph
117     if (n[0] < 0 && n[1] >= 0) {
118         ph = Math.PI + Math.atan(n[1] / n[0]);

```

B Java-Quelltexte

```

119     }
120
121     //3. Quadrant  $\phi = 180 + \phi$ 
122     if (n[0] < 0 && n[1] < 0) {
123         ph = Math.PI + Math.atan(n[1] / n[0]);
124     }
125
126     //4. Quadrant  $\phi = 360 + \phi$ 
127     if (n[0] > 0 && n[1] < 0) {
128         ph = 2 * Math.PI + Math.atan(n[1] / n[0]);
129     }
130 }
131
132 //Umrechnung in Horizontsystem: Azimut, Zenit
133 az = 0;
134 az = 270.0 - Math.toDegrees(ph);
135 if (az < 0) {
136     az = az + 360.0;
137 }
138 if (az >= 360) {
139     az = az - 360.0;
140 }
141 else {}
142 r += " " + az + "\t";
143 r += " " + h + "\t";
144 rep += "Azimut:␣" + az;
145 az = Math.toRadians(az);
146
147 //Umrechnung in festes Aequatorialsystem
148 //Sternzeit
149 double star, starloc, jd0;
150 jd0 = Math.floor(365.25 * (Y + 4716)) + Math.floor(30.6001 * (M + 1)) +
151     D - 13 - 1524.5;
152 star = 6.664520 + 0.0657098244 * (jd0 - 2451544.5) + 1.0027379093 * utc;
153 star = star - Math.floor(star / 24) * 24;
154 starloc = star + gl / 15;
155
156 //Deklination, Stundenwinkel, Rektaszension
157 double dek, stdc, stds, std, rek;
158 dek = Math.asin(Math.cos(Math.toRadians(90 - h)) *
159     Math.sin(Math.toRadians(gb)) - Math.sin(Math.toRadians(90 - h)) *
160     Math.cos(az) * Math.cos(Math.toRadians(gb)));
161 stdc = Math.cos(Math.toRadians(90 - h)) * Math.cos(Math.toRadians(gb)) -
162     Math.sin(Math.toRadians(90 - h)) * Math.cos(az) *
163     Math.sin(Math.toRadians(gb)) / Math.cos(dek);
164
165 stds = Math.sin(Math.toRadians(90 - h)) * Math.sin(az) / Math.cos(dek);
166 std = 0;
167 if (stdc == 0 && stds > 0) {
168     std = Math.PI / 2;
169 }
170 if (stdc == 0 && stds <= 0) {
171     std = -Math.PI / 2;
172 }
173 if (stdc != 0) {
174
175     //1. Quadrant  $\sigma = \sigma$ 
176     if (stdc > 0 && stds >= 0) {
177         std = Math.atan(stds / stdc);

```

B Java-Quelltexte

```
178     }
179
180     //2. Quadrant std = 180 + std
181     if (stdc < 0 && stds >= 0) {
182         std = Math.PI + Math.atan(stds / stdc);
183     }
184
185     //3. Quadrant std = 180 + std
186     if (stdc < 0 && stds < 0) {
187         std = Math.PI + Math.atan(stds / stdc);
188     }
189
190     //4. Quadrant std = 360 + std
191     if (stdc > 0 && stds < 0) {
192         std = 2 * Math.PI + Math.atan(stds / stdc);
193     }
194 }
195 if (std < 0) {
196     std = std + 2 * Math.PI;
197 }
198 else {}
199 std = Math.toDegrees(std) * 24 / 360;
200 rep += "\nStdwinkel: " + std;
201 rek = starloc - std;
202 if (rek < 0) {
203     rek = rek + 24;
204 }
205 else {}
206 r += rek;
207 rep += "\nRektaszension: " + rek;
208 dek = Math.toDegrees(dek);
209 r += "\t" + dek;
210 rep += "\nDeklination: " + dek;
211
212 //Bericht ausgeben
213 if (report == true) {
214     System.out.println(rep);
215 }
216 else {}
217 }
218 return r;
219 }
220
221 //Betrag eines 3D-Vektors
222 public static double length(double v[], double w[])
223 {
224     return Math.sqrt(sq(v[0] - w[0]) + sq(v[1] - w[1]) + sq(v[2] - w[2]));
225 }
226
227 //Quadrat einer Zahl
228 public static double sq(double v)
229 {
230     return v*v;
231 }
232
233 //Kreuzprodukt zweier Vektoren v x w
234 public static double[] kreuz(double v[], double w[])
235 {
236     double r[] = new double[3];
```

```

237     r[0] = v[1] * w[2] - v[2] * w[1];
238     r[1] = v[2] * w[0] - v[0] * w[2];
239     r[2] = v[0] * w[1] - v[1] * w[0];
240     return r;
241 }
242
243 //Differenz zweier 3D-Vektoren v - w
244 public static double[] diff(double v[], double w[])
245 {
246     double r[] = new double[3];
247     r[0] = v[0] - w[0];
248     r[1] = v[1] - w[1];
249     r[2] = v[2] - w[2];
250     return r;
251 }
252
253 //Matrizenmultiplikation 3x3
254 public static double[] mm(double m[][], double v[])
255 {
256     double r[] = new double[3];
257     for (int i = 0; i <= 2; i++) {
258         r[i] = m[i][0] * v[0] + m[i][1] * v[1] + m[i][2] * v[2];
259     }
260     return r;
261 }
262
263 //Normierung eines 3D-Vektors
264 public static double[] norm(double v[])
265 {
266     double r[] = new double[3];
267     r[0] = v[0] / Math.sqrt(sq(v[0]) + sq(v[1]) + sq(v[2]));
268     r[1] = v[1] / Math.sqrt(sq(v[0]) + sq(v[1]) + sq(v[2]));
269     r[2] = v[2] / Math.sqrt(sq(v[0]) + sq(v[1]) + sq(v[2]));
270     return r;
271 }
272 }

```

B.11 anglet

```

1 import java.io.*;
2
3 public class anglet
4 {
5     public static void main(String[] args)
6     {
7
8         //Objekt der Klasse angle initialisieren
9         angle test = new angle();
10        if (args.length != 3) {
11            System.out.println("Benutzung: _java_ angle_<hits>_<inputfile>_<outputfile>");
12            System.exit(1);
13        }
14
15        //Detektorkoordinaten aus config.txt
16        double det[][] = new double[4][3];
17        double eg = 0;
18        try {

```


B Java-Quelltexte

```
19 LineNumberReader f = new LineNumberReader(new FileReader("config.txt"));
20 String line;
21 while ((line = f.readLine()) != null) {
22     for (int i = 0; i <= 3; i++) {
23         for (int j = 0; j <= 2; j++) {
24             det[i][j] = Double.parseDouble(line.substring(0, line.indexOf("\n")));
25             line = line.substring(line.indexOf("\n") + 1);
26             System.out.println("\n" + (1 + i) + "\n" + (1 + j) + "\n:\n" + det[i][j]);
27         }
28     }
29     test.gb = Double.parseDouble(line.substring(0, line.indexOf("\n")));
30     System.out.println("Geographische_Breite_[" + "\u00b0":\n" + test.gb);
31     line = line.substring(line.indexOf("\n") + 1);
32     test.gl = Double.parseDouble(line);
33     System.out.println("Geographische_Laenge_[" + "\u00b0":\n" + test.gl);
34     for (int i = 0; i <= 3; i++) {
35         for (int j = 0; j <= 3; j++) {
36             if (test.length(det[i], det[j]) >= eg) {
37                 eg = test.length(det[i], det[j]);
38             }
39             else {}
40         }
41     }
42
43     //Event-Gate
44     eg = eg / 2.99792458E-1;
45     System.out.println("EventGate_[" + ns]:\n" + eg);
46     f.close();
47     break;
48 }
49 } catch (IOException e) {
50     System.out.println("Fehler_beim_Lesen_der_Datei!");
51 }
52
53 //Daten einlesen und ausgeben
54 String inline, input, output;
55 input = args[1];
56 output = args[2];
57 int hits = Integer.parseInt(args[0]);
58 Long current = 0l;
59 boolean trig = false;
60 String[] buffer = new String[20];
61 int c = 0;
62 try {
63     LineNumberReader f = new LineNumberReader(new FileReader(input));
64     FileWriter ou = new FileWriter(output);
65     while ((inline = f.readLine()) != null) {
66         Long time = Long.parseLong(inline.substring(0, inline.indexOf("\n")));
67         if (trig == true && (((double) (time - current)) / 1000) <= eg) {
68             c++;
69             buffer[c] = inline;
70         }
71         if (trig == false) {
72             f.mark(1000);
73             current = time;
74             trig = true;
75             c = 0;
76             buffer[c] = inline;
77         }
78     }
79 }
```

```

78         if (trig == true && (((double) (time - current)) / 1000) > eg) {
79             if (c + 1 >= hits) {
80                 System.out.println("Schreibe...");
81                 outprint(test, ou, buffer, c, det);
82             }
83             f.reset();
84             trig = false;
85         }
86     }
87     f.close();
88     ou.close();
89 } catch (IOException e) {
90     System.out.println("Fehler beim Lesen der Datei!");
91 }
92 }
93
94 //Methode zum Ausgabe der berechneten Richtungen
95 public static void outprint(angle winkel, FileWriter out, String[] data,
96                             int len, double[][] detekt)
97 {
98     double x[][] = new double[3][3];
99     long t[] = new long[3];
100    double zeit;
101    String datum;
102    if ((len + 1) == 3) {
103        for (int i = 0; i <= 2; i++) {
104            t[i] = Long.parseLong(data[i].substring(0, data[i].indexOf("_")));
105            System.out.println("Zeiten:_ " + t[i]);
106            x[i] = detekt[Integer.parseInt(data[i].substring(
107                data[i].lastIndexOf("_") + 1))];
108            System.out.println("Kanal:_ " + Integer.parseInt(data[i].substring(
109                data[i].lastIndexOf("_") + 1)));
110        }
111        zeit = t[0] * 10E-13 / 3600d;
112        System.out.println("Zeit:_ " + zeit);
113        datum = data[0].substring(data[0].lastIndexOf("_") - 6,
114                                   data[0].lastIndexOf("_"));
115        System.out.println("Datum:_ " + datum);
116        try {
117            out.write(winkel.angles(x[0], x[1], x[2], t, zeit, datum, true) + "\r\n");
118        } catch (IOException e) {
119            System.out.println("Fehler beim Lesen der Datei!");
120        }
121    }
122    if ((len + 1) > 3) {
123        System.out.println("Hits:_ " + (len + 1));
124    }
125 }
126 }

```

B.12 histo

```

1  /*
2   * histo
3   */
4
5  import java.io.*;

```

B Java-Quelltexte

```
6 import java.text.*;
7
8 public class histo
9 {
10     public static void main(String[] args)
11     {
12         if (args.length != 8) {
13             System.out.println("Benutzung: java histo <xtics> <xmin> <xmax>"
14                               + " <ytics> <ymin> <ymax> <input file> <output file>");
15             System.exit(1);
16         }
17
18         //Kommandozeile einlesen
19         double xtics = Double.parseDouble(args[0]);
20         double xmin = Double.parseDouble(args[1]);
21         double xmax = Double.parseDouble(args[2]);
22         double ytics = Double.parseDouble(args[3]);
23         double ymin = Double.parseDouble(args[4]);
24         double ymax = Double.parseDouble(args[5]);
25         String input = args[6];
26         String output = args[7];
27
28         //Histogramm-Array aufstellen
29         int x = (int) (Math.ceil((xmax - xmin) / xtics));
30         int y = (int) (Math.ceil((ymax - ymin) / ytics));
31         int[][] counter = new int[x][y];
32
33         try {
34
35             //Streams initialisieren
36             LineNumberReader f = new LineNumberReader(new FileReader(input));
37             String line;
38             int lines = 0, empty = 0;
39
40             //Daten einlesen
41             while ((line = f.readLine()) != null) {
42
43                 //Leerzeilen
44                 if (line.length() == 0) {
45                     empty++;
46                 }
47                 //Datenzeilen zerlegen und sortieren
48                 else {
49                     line = line.substring(line.indexOf("\t") + 1, line.length());
50                     line = line.substring(line.indexOf("\t") + 1, line.length());
51                     double rekt = Double.parseDouble(line.substring(0, line.indexOf("\t")));
52                     line = line.substring(line.indexOf("\t") + 1, line.length());
53                     double dek = Double.parseDouble(line);
54                     for (int i = 0; i < x; i++) {
55                         for (int l = 0; l < y; l++) {
56                             if ((xmin + (double) (i) * xtics) <= rekt &&
57                                 rekt < (xmin + (double) (i + 1) * xtics) &&
58                                 (ymin + (double) (l) * ytics) <= dek &&
59                                 dek < (ymin + (double) (l + 1) * ytics)) {
60                                 counter[i][l]++;
61                             }
62                             else {}
63                         }
64                     }
65                 }
66             }
67         }
68     }
69 }
```

B Java-Quelltexte

```
65     }
66     lines++;
67 }
68 f.close();
69
70 //Intervalle und Treffer in Ausgabedatei schreiben
71 FileWriter ou = new FileWriter(output);
72 DecimalFormat form = new DecimalFormat("#0.00");
73 for (int i = 0; i < x; i++) {
74     for (int l = 0; l < y; l++) {
75         ou.write(" " + form.format(xmin + (double) (i + 1) * xtics) + "\t" +
76                 form.format(ymin + (double) (l + 1) * ytics) + "\t" +
77                 counter[i][l] + "\r\n");
78     }
79 }
80 ou.close();
81 System.out.println("Leere_Zeilen:_ " + empty);
82 System.out.println("Zeilen_gesamt:_ " + lines);
83 } catch (IOException e) {
84     System.out.println("Fehler_beim_Lesen_der_Datei!");
85 }
86 }
87 }
```

Literaturverzeichnis

- [B⁺06] J. Bluemer et al. *Astroteilchenphysik in Deutschland – Ein Forschungsgebiet im Aufbruch*, 2006. <http://www.astroteilchenphysik.de>.
- [Ber] H.-G. Berns. *QNet2 DAQ card tips'n'tricks*. <http://neutrino.phys.washington.edu/~berns/WALTA/Qnet2/manuals/tipsandtricks.html>.
- [Elj] Eljen Technology. *EJ 200 Datenblatt*. <http://www.eljentechnology.com/datasheets/EJ200%20data%20sheet.pdf>.
- [Fer] Fermilab. *Cosmic Ray e-Lab*. <http://www18.i2u2.org/elab/cosmic/home/>.
- [FIB] FIBS. *Lehrerfortbildung in Bayern*. <http://www.km.bayern.de/km/lehrerbildung/lehrerfortbildung/>.
- [Fri09] Friedrich-Alexander-Universität Erlangen-Nürnberg. *Studierendenzahlen der Studiengänge nach Lehreinheiten - Lehreinheit: Physik (Nat. Fak.)*, 2009. <http://www.uni-erlangen.de/universitaet/statistik/studierende/lehreinheiten/physik.pdf>.
- [Fry10] S. Fryska. *Simulationsuntersuchungen zur Messung der kosmischen Strahlung mit Szintillationsdetektoren*, Bachelorarbeit, Universität Erlangen-Nürnberg, 2010.
- [Grea] H.-C. Greier. *Berechnung der Julianischen Tagzahl JD*. <http://www.greier-greiner.at/hc/bsp/juldat/index.htm>.
- [Greb] H.-C. Greier. *Berechnung der Ortssternzeit*. http://www.greier-greiner.at/hc/bsp/sid_time/index.htm.
- [Grec] H.-C. Greier. *Nautisches Dreieck*. http://www.greier-greiner.at/hc/sph_dreieck.htm.
- [Heb01] T. Hebbeker. *Remarks on SKYVIEW grid size*, 2001. <http://www.physik.rwth-aachen.de/~hebbeker/skyview.ps>.
- [jav] java-forum.org. *Sourcecode des Programms comframe*. <http://www.java-forum.org/java-basics-anfaenger-themen/38471-kartenleser-ueber-serielle-schnittstelle-auslesen.html>.

- [Mer08] G. Merzyn. *Naturwissenschaften Mathematik Technik – immer unbeliebter?*. Schneider-Verlag, 2008.
- [Mer09] G. Merzyn. Polarisierender Physikunterricht. *Physik in unserer Zeit*, (6):312–313, 2009.
- [MR08] G. Manhardt and H. Ruch. *Die Seminare in der gymnasialen Oberstufe*. Kastner Verlag, 2008.
- [Par08] Particle Data Group. *Particle Physics Booklet*, July 2008. <http://pdg.lbl.gov>.
- [RJPB10] J. Rylander, T. Jordan, J. Paschke, and H.-G. Berns. *QuarkNet Cosmic Ray Muon Detector User’s Manual Series ‘6000’ DAQ*, January 2010. <http://www18.i2u2.org/elab/cosmic/teacher/library/upload/b/ba/6000CRMDUserManual.pdf>.
- [Ros52] B. Rossi. *High energy particles*. Prentice Hall, New York, 1952.
- [SBS10] F. Sheidaei, M. Bahmanabadi, and J. Samimi. Moon shadow on high energy cosmic ray in data of a small extensive air shower array. *Astroparticle Physics*, (33):330–334, 2010.
- [Sena] Sens-Tech. *P30CW5 Datenblatt*. http://www.sens-tech.com/assets/media/files/Data%20Sheets/P30CW5_iss01.pdf.
- [Senb] Sens-Tech. *P30CW5 technische Zeichnung*. <http://www.sens-tech.com/assets/media/files/Drawings/25894c1350-02s1.pdf>.
- [Sil] Silabs. *CP210x-Chip Treiber*. <https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>.
- [SL07] P. Solis and L. Liu. *The Speed and Lifetime of Cosmic Ray Muons*, Praktikumsreport, Massachusetts Institute of Technology, 2007. <http://web.mit.edu/lululiu/Public/pixx/not-pixx/muons.pdf>.
- [Wika] Wikipedia. *Drehmatrix*. <http://de.wikipedia.org/wiki/Drehmatrix>.
- [Wikb] Wikipedia. *Raumwinkel*. <http://de.wikipedia.org/wiki/Raumwinkel>.

Danksagung

„Keine Schuld ist dringender, als die, Dank zu sagen.“

Marcus Tullius Cicero

In diesem Sinne möchte ich mich herzlich bei der Erlanger HESS-Gruppe mit Prof. Christian Stegmann an der Spitze bedanken, die es mir ermöglicht hat, zu dieser interessanten Thematik eine Arbeit anzufertigen. Mein ganz besonderer Dank gilt Kathrin Valerius. Sie hat sehr viel Geduld für mich aufgebracht und mich stets mit ihren Ideen unterstützt und motiviert. An dieser Stelle möchte ich ebenfalls Rezo Shanidze erwähnen, der mir mit seinen Anregungen und Skripten zur Datenauswertung eine große Hilfe war. Mein Dank gilt auch Friedrich Stinzing, der mir bei der Montage der ersten Detektoren sehr geholfen hat. Danke auch an Bernhard Glück für die Einrichtung meines Labor-Rechners. Außerdem ist hier Johannes Montag zu nennen, der uns mit dem CRMD-Set in seinen Vorbereitungsräumen großzügig Asyl gewährte. Für die Bereitstellung ihrer Kamera möchte ich mich hiermit auch bei Edith Avram bedanken.

Wahrheitsgemäße Erklärung

Ich erkläre hiermit, dass ich

- die vorliegende Arbeit selbständig und ohne unerlaubte Hilfe angefertigt habe,
- außer den im Schrifttumsverzeichnis angegebenen Quellen und Hilfsmitteln keine weiteren benutzt und alle Stellen, die aus dem Schrifttum ganz oder annähernd entnommen sind, als solche kenntlich gemacht und einzeln nach ihrer Herkunft unter Bezeichnung der Ausgabe (Auflage und Jahr des Erscheinens), des Bandes des benützten Werkes in der Zulassungsarbeit nachgewiesen habe,
- die Zulassungsarbeit noch keiner anderen Stelle zur Prüfung vorgelegt habe.

Erlangen, den 24.09.2010

Friedrich Meingast