

Fortgeschrittenen Praktikum I
Kurze Halbwertszeiten

Wiebke Herzberg und Kolja Glogowski

15. Oktober 2004

Inhaltsverzeichnis

1. Aufgabenstellung	3
2. Theoretische Grundlagen	3
2.1. Methode der verzögerten Koinzidenz	3
2.2. Die Slope Methode	3
2.3. Funktionsweise der verwendeten Geräte	5
3. Versuchsaufbau und Durchführung	7
3.1. Kalibrierung des TAC	7
3.2. Aufnahme der Energiespektren	8
3.3. Einstellung des Diskriminator-Fenster	11
3.4. Aufnahme des Zeitspektrums	14
4. Auswertung	17
4.1. Zeitkalibrierung des TAC	17
4.2. Bestimmung der mittleren Lebensdauer und der Halbwertszeit	18
5. Zusammenfassung	20
A. Quelltexte	21
A.1. Kalibrierung des TAC (<code>tac_kalib.py</code>)	21
A.2. Erzeugung der Histogramme aus den Messdaten (<code>messung.py</code>)	22
A.3. Bestimmung der Lebensdauer und der Halbwertszeit (<code>lebensdauer.py</code>)	24
A.4. Anzeigen der Histogramme (<code>showhistos.py</code>)	26

1. Aufgabenstellung

1. Aufnahme der Energiespektren der γ -Strahlung von ^{22}Na und ^{57}Co .
2. Bestimmung der Halbwertszeit und der mittleren Lebensdauer des $14,4\text{ keV}$ -Zustands von ^{57}Fe .

2. Theoretische Grundlagen

^{57}Co hat eine Halbwertszeit von 270 Tagen. Es zerfällt zu 99,8% durch Elektroneneinfang (EC) zum $136,3\text{ keV}$ -Zustand von ^{57}Fe . Dieser Zustand geht zu 98% unter Aussendung eines $121,9\text{ keV}$ γ -Quants in den $14,4\text{ keV}$ -Zustand über. Die restlichen 11% gehen unter Aussendung eines $136,3\text{ keV}$ γ -Quants direkt in den Grundzustand über. Beim Übergang des $14,4\text{ keV}$ -Zustands in den Grundzustand kommt es in einem Verhältnis von 10 : 1 zu Innerer Konversion, also dem Übertragen der Energie auf ein Hüllenelektron, bzw. dem Emitieren eines $14,4\text{ keV}$ γ -Quants.

2.1. Methode der verzögerten Koinzidenz

Die Bestimmung der Halbwertszeit und der mittleren Lebensdauer des $14,4\text{ keV}$ -Zustands lässt sich durch die *Methode der verzögerten Koinzidenz* realisieren. Dies ist eine oft angewandte Methode zur Messung von Halbwertszeiten sehr kurzlebiger Zustände.

Da der $14,4\text{ keV}$ -Zustand aus dem $136,3\text{ keV}$ -Zustand hervorgeht zeigt ein 122 keV -Signal den Existenzbeginn des $14,4\text{ keV}$ -Zustands an. Während ein $14,4\text{ keV}$ -Signal den Zerfall dieses Zustands, also das Ende der Lebensdauer anzeigt. Daher müsste man, um die Lebensdauer zu bestimmen einfach nur die Zeit zwischen diesen beiden Pulsen messen. Da der $14,4\text{ keV}$ -Zustand aber nicht nur durch Aussenden eines γ -Quants sondern auch durch innere Konversion zerfällt, würde man nach vielen 122 keV -Signalen gar kein $14,4\text{ keV}$ -Signal erhalten, also zu vielen Startsignalen (in zehn von elf Fällen!) kein zugehöriges Stoppsignal. Daher beginnen wir unsere Messung immer mit dem $14,4\text{ keV}$ -Signal und verzögern das eigentlich vorangegangene dazugehörige 122 keV Signal mittels des variablen Delays solange, dass es erst nach dem $14,4\text{ keV}$ -Signal eintrifft und somit als Stoppsignal dienen kann. Die Zeit zwischen beiden Signalen wird mit dem TAC gemessen.

2.2. Die Slope Methode

Beim radioaktiven Zerfall handelt es sich um einen statistischen Prozess. Für die Wahrscheinlichkeitsdichte $f(t)$ eines radioaktiven Zerfallsereignisses gilt:

$$f(t) = \begin{cases} \lambda e^{-\lambda t} & \text{für } t \geq 0 \\ 0 & \text{für } t < 0 \end{cases} \quad (1)$$

Echt koinzidente Ereignisse würden jeweils im Zeitspektrum eine scharfe Linie ergeben und die aufgenommene Kurve würde der Verteilung $f(t)$ folgen. In Wirklichkeit ist aber jedes Ereignis gaußförmig verschmiert, die zeitliche Verteilung von echt koinzidenten Ereignissen folgt also der sogenannten *prompten Kurve* $P(t)$. Daher handelt es sich bei der aufgenommenen Kurve $F(t)$ um eine Faltung von $f(t)$ und $P(t)$:

$$F(t) = (f \otimes P)(t) = \int_0^{\infty} f(t') P(t - t') dt' \quad (2)$$

Durch die Substitution $z = t - t'$ erhält man daraus:

$$\begin{aligned} F(t) &= - \int_t^{-\infty} f(t - z) P(z) dz = \int_{-\infty}^t f(t - z) P(z) dz \\ &= \int_{-\infty}^t \lambda e^{-\lambda(t-z)} P(z) dz = \lambda e^{-\lambda t} \int_{-\infty}^t e^{\lambda z} P(z) dz \end{aligned} \quad (3)$$

Daraus ergibt sich für die zeitliche Ableitung:

$$\begin{aligned} \frac{dF(t)}{dt} &= -\lambda^2 e^{-\lambda t} \int_{-\infty}^t e^{\lambda z} P(z) dz + \lambda e^{-\lambda t} e^{\lambda t} P(t) \\ &= -\lambda F(t) + \lambda P(t) = \lambda (P(t) - F(t)) \end{aligned} \quad (4)$$

Für das Maximum von $F(t)$ gilt nun:

$$\frac{dF(t)}{dt} = 0 \quad \Rightarrow \quad P(t) = F(t) ,$$

also liegt es am selben Ort wie das Maximum der prompten Kurve. Betrachtet man nun den Fall $F(t) \gg P(t)$, so ergibt sich mit (4):

$$\frac{dF(t)}{dt} = \lambda (P(t) - F(t)) \approx -\lambda F(t) \quad \Rightarrow \quad F(t) \approx A e^{-\lambda t} . \quad (5)$$

Die Zerfallskonstante λ lässt sich also in Bereichen, in denen $P(t)$ viel kleiner als $F(t)$ ist, aus dem Exponenten einer exponentiellen Fitfunktion bestimmen, oder bei halblogarithmischer Auftragung aus der Steigung (*engl: slope*) der Regressionsgeraden. Aufgrund des gaußförmigen Verlaufs der prompten Kurve $P(t)$, lässt sich der Bereich in dem sie noch einen relevanten Beitrag zur Messkurve $F(t)$ liefert abschätzen, indem man die Flanke der Kurve am Maximum spiegelt.

2.3. Funktionsweise der verwendeten Geräte

Der Detektor

Der Detektor besteht aus einem Szintillator, einem Photomultiplier und einem Vorverstärker.

Ein Szintillator besteht aus einem Kristall, der häufig auch mit Fremdatomen dotiert ist. Die einfallenden γ -Quanten geben ihre Energie an die Elektronen des Kristalls ab, die dadurch in das Leitungsband gehoben werden und an ihrer Stelle ein Loch hinterlassen. Rekombinieren diese Elektron-Loch-Paare, z.B. an durch die Fremdatome verursachten Fehlstellen, entstehen Lichtblitze (Photonen) im nahen UV-Bereich. Je größer die Energie des einfallenden γ -Quants desto mehr Elektronen werden herausgelöst und desto mehr Photonen entstehen daher auch bei der Rekombination. Diese Photonen treffen nun auf die Kathode des Photomultipliers wo sie wiederum Elektronen herauslösen. Die Elektronen werden innerhalb des Photomultipliers durch eine Spannung von Dynode zu Dynode beschleunigt, wo sie immer mehr zusätzliche Elektronen herauslösen. Das Signal wird dadurch soweit verstärkt, dass man schließlich, durch die an der Anode abfließenden Elektronen, einen messbaren Impuls erhält. Dieser gelangt dann zum Vorverstärker und wird dort noch ein weiteres Mal verstärkt.

Der Einkanaldiskriminator (SCA)

Mit dem Einkanaldiskriminator lässt sich der Energiebereich einschränken aus dem ein Signal weitergeleitet wird. Man kann sozusagen ein Fenster auf einen bestimmten Energiebereich setzen, und der SCA sendet nur einen Puls aus, wenn das Eingangssignal in diesem Energiebereich liegt. Alle anderen Eingangspulse werden verworfen.

Der Multichannel Analyzer (MCA)

Der Multichannel Analyser, in unserem Fall ein Computer, hat einen Wertebereich von 10 Volt, der in 1024 Kanäle eingeteilt ist. Je nach Höhe der Eingangsamplitude wird das Signal einem der Kanäle zugeordnet.

Der Zeit-Impulshöhen-Konverter (TAC)

Der Zeit-Impulshöhen-Konverter (TAC) wandelt Zeitintervalle in dazu proportionale Spannungsimpulse um. Dazu wird bei einem Startimpuls ein konstanter Strom eingeschaltet, der zur Ladung eines Kondensators führt. Tritt nun ein Stoppimpuls auf, so wird der so aufintegrierte Strom in einen Spannungsimpuls von fest definierter zeitlicher

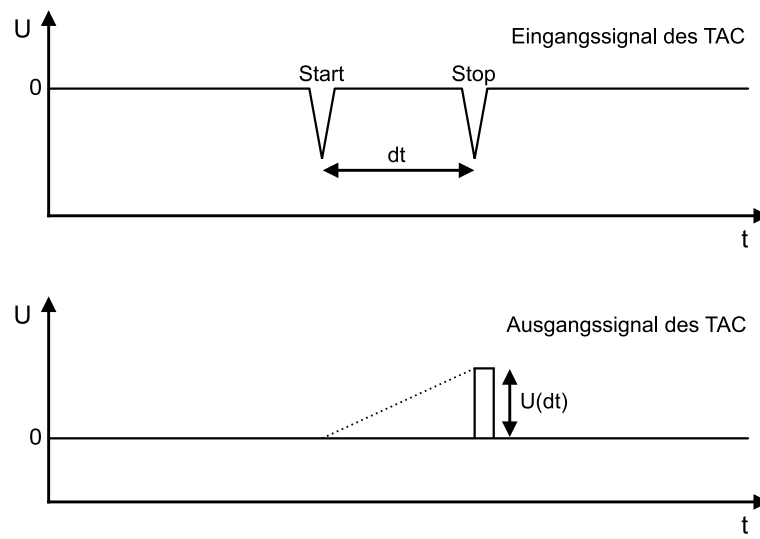


Abbildung 1: Funktionsweise des Zeit-Impulshöhen-Konverters (TAC)

Länge umgewandelt, dessen Höhe proportional zum Zeitintervall zwischen Start- und Stoppimpuls ist (siehe Abbildung 1).

3. Versuchsaufbau und Durchführung

Die Detektoren (bestehend aus Szintillator, Photomultiplier und Vorverstärker) betrieben wir mit einer konstanten Versorgungsspannung von 1117 V am linken Detektor (Detektor 1) und 607 V am rechten Detektor (Detektor 2). Zu Beginn der Messung hatten wir Schwierigkeiten ein Signal am rechten Detektor zu messen. Das lag daran, dass dieser am Abend vorher durch einen anderen Detektor ausgetauscht worden war, der, so wie der linke, auch eine Versorgungsspannung von über 1000 V benötigt. Da sich der neue Detektor jedoch zur Aufnahme des $14,4\text{ keV}$ -Peaks nicht besonders gut eignete, tauschten wir ihn wieder durch den ursprünglichen Detektor aus.

3.1. Kalibrierung des TAC

Zur Kalibrierung des TAC verwendeten wir eine ^{22}Na Quelle, bei der es sich um einen β^+ -Strahler handelt. Ursprünglich sollten die beiden 511 keV γ -Quanten, die bei der Annihilation des emittierten Positrons mit einem Elektron entstehen und gleichzeitig in entgegengesetzter Richtung ausgesendet werden, dazu verwendet werden den TAC zu kalibrieren. Stattdessen verwendeten wir eine einfachere Methode (siehe Abbildung 2):

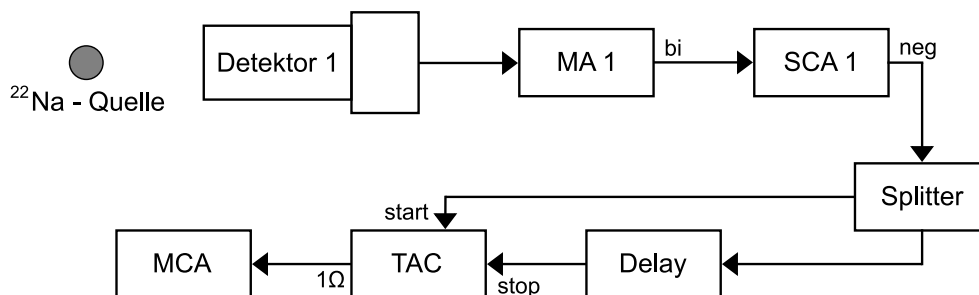


Abbildung 2: Schaltung zur Kalibrierung des TAC

Wir nahmen mit einem der Detektoren nur eines der beiden γ -Quanten auf und teilten das Signal mit einem Splitter in zwei Signale auf. Das eine der beiden Signale gaben wir direkt auf den Start-Eingang des TAC, während wir das andere vorher über eine Kaskade geeichter BNC-Kabel verzögerten und danach an den Stopp-Eingang des TAC weitergaben. Nun variierten wir die Länge des Kabels und damit die Verzögerung, die das eine der beiden Signale zu durchlaufen hatte und nahmen dabei das Signal des TAC mit dem Multichannel-Analyser (MCA) auf. Das Fenster des SCA öffneten wir ganz und verwendeten folgende Einstellungen am MA:

Gain	Coarse gain	Shaping time
5,762	20	0,5

Das Messergebnis ist in Abbildung 3 zu sehen.

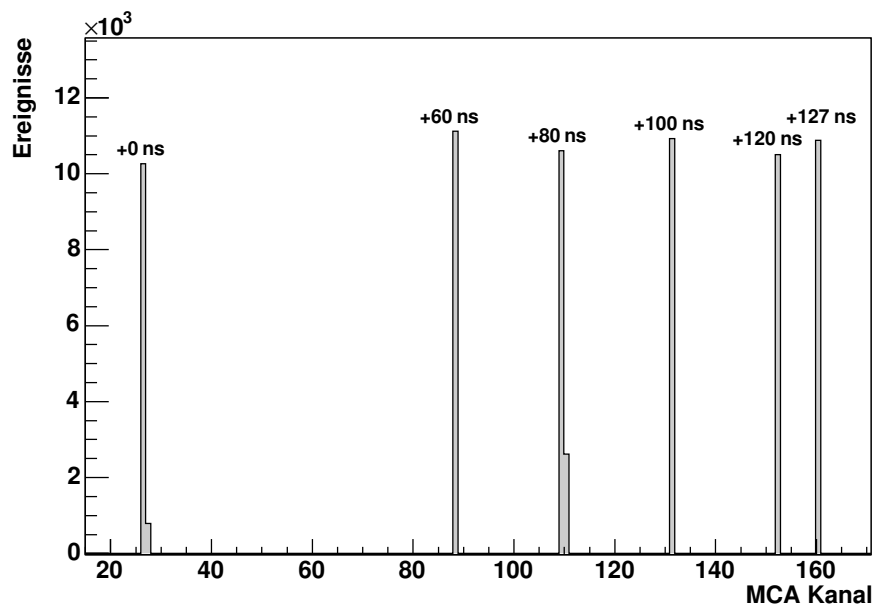


Abbildung 3: Messung zur Kalibrierung des TAC

3.2. Aufnahme der Energiespektren

Zur Aufnahme der jeweiligen Energiespektren griffen wir das Signal am Hauptverstärker (MA) ab und führten direkt zum MCA (siehe Abbildung 4). Dabei achteten wir mit Hilfe des Oszilloskops darauf, dass wir den Spannungsbereich des MCA von etwa 5 V nicht überschritten.

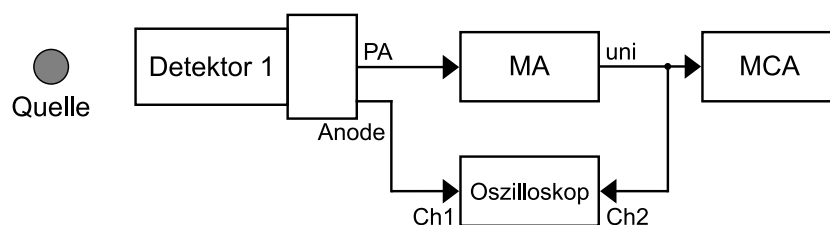


Abbildung 4: Schaltung zur Aufnahme der Energiespektren

Wir verwendeten für die Aufnahme der Spektren folgende Einstellungen am MA:

Spektrum	Detektor	Gain	Coarse gain	Shaping time
Cobalt	1	5,74	100	0,5
Cobalt	2	7,22	500	0,5
Natrium	1	2,20	50	0,5

Abbildung 5 zeigt die Aufnahme des ^{22}Na -Spektrums, dabei ist der 511 keV -Peak auf der rechten Seite deutlich zu erkennen. Abbildungen 6 und 7 zeigen das mit unterschiedlichen Detektoren aufgenommene ^{57}Co -Spektrum. Wie man sieht kann man den 14,4 keV -Peak mit Detektor 1 nicht erkennen, während er mit Detektor 2 sehr gut zu sehen ist.

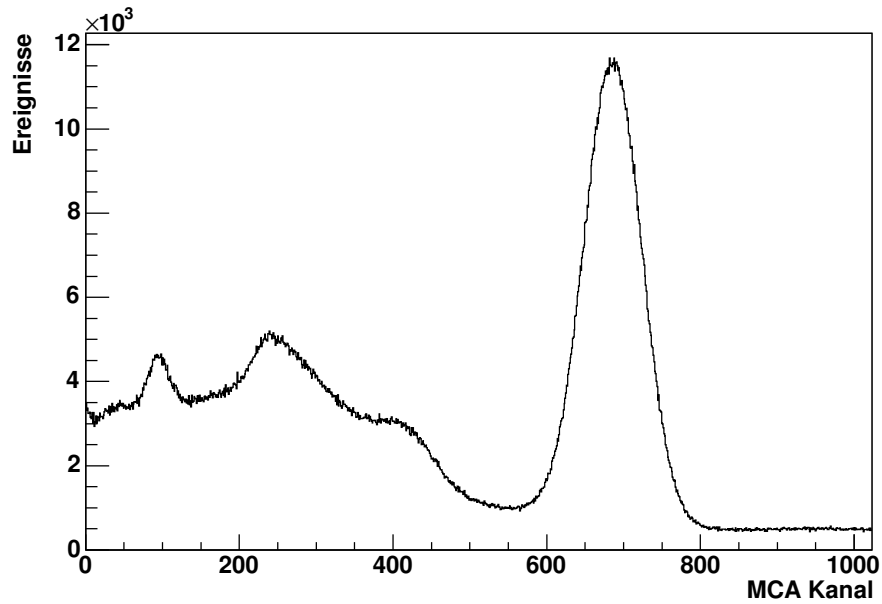


Abbildung 5: Energiespektrum von ^{22}Na , aufgenommen mit Detektor 1

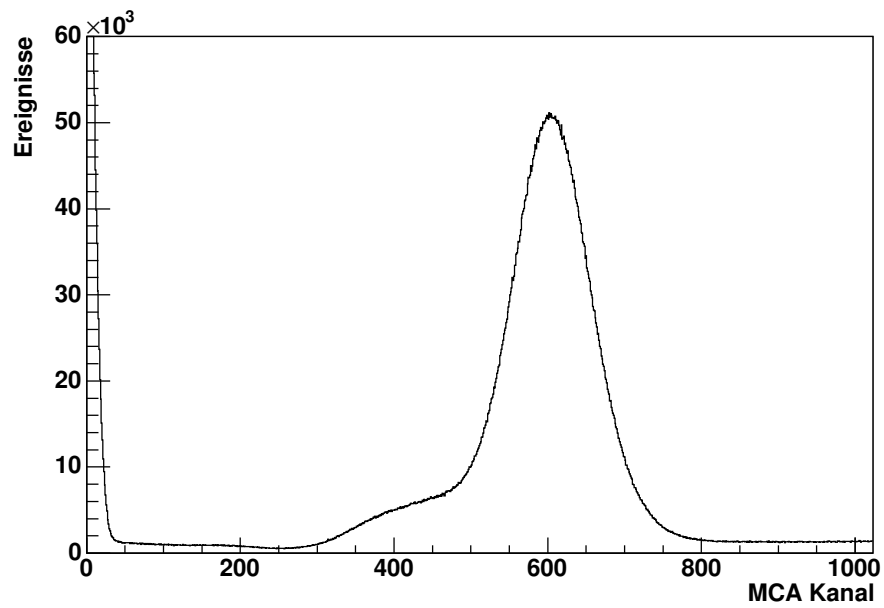


Abbildung 6: Energiespektrum von ^{57}Co , aufgenommen mit Detektor 1

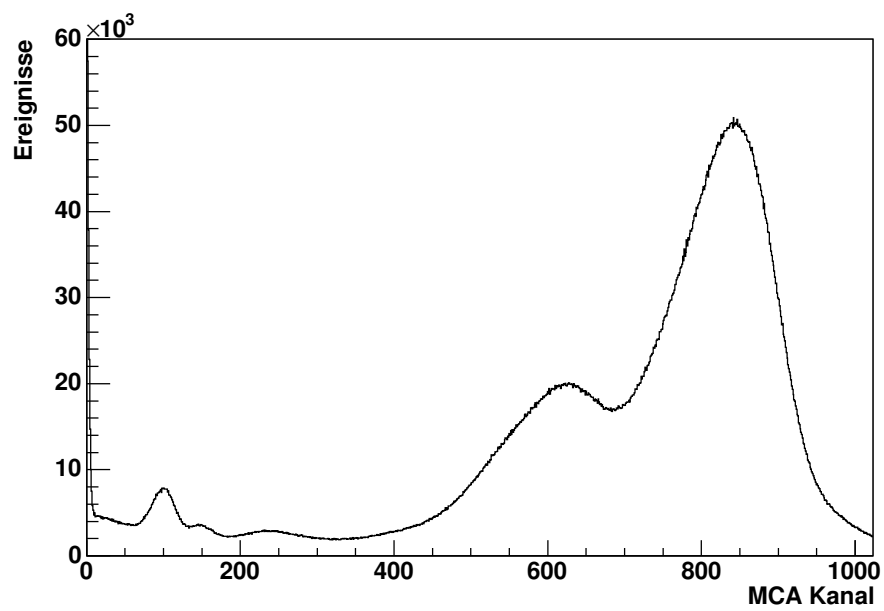


Abbildung 7: Energiespektrum von ^{57}Co , aufgenommen mit Detektor 2

3.3. Einstellung des Diskriminator-Fenster

Um für unsere Messung auch wirklich nur die γ -Quanten der gewünschten Energie, nämlich mit dem einen Detektor 122 keV und mit dem anderen $14,4\text{ keV}$, als Signal zu erhalten benutzen wir den SCA um die Energiefenster einzustellen. Zum Auffinden der 122 keV -bzw. $14,4\text{ keV}$ -Peaks nutzten wir das mit dem jeweiligen Szintillator aufgenommene Spektrum und stellten dann durch Variation der oberen bzw. unteren Fensterkante das Energiefenster genau auf die Breite der Peaks ein, sodass statt dem gesamten Spektrum nur noch Signale, welche zu diesen Peaks gehörten, gezählt wurden. Abbildungen 7 und 8 zeigen die beiden Schaltungen für die Detektoren bei der Einstellung ihrer zugehörigen Fenster.

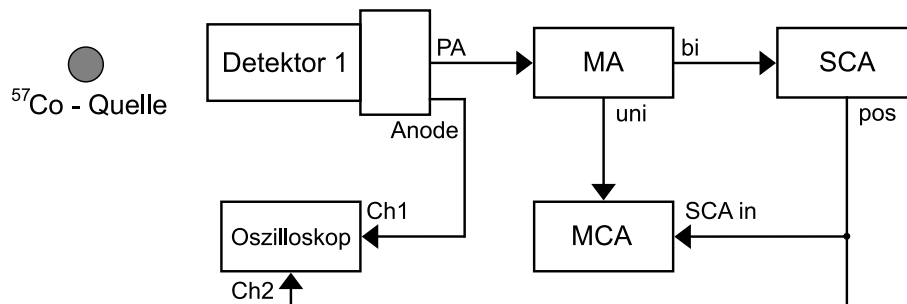


Abbildung 8: Schaltung zur Einstellung des Diskriminators auf das 122 keV -Fenster

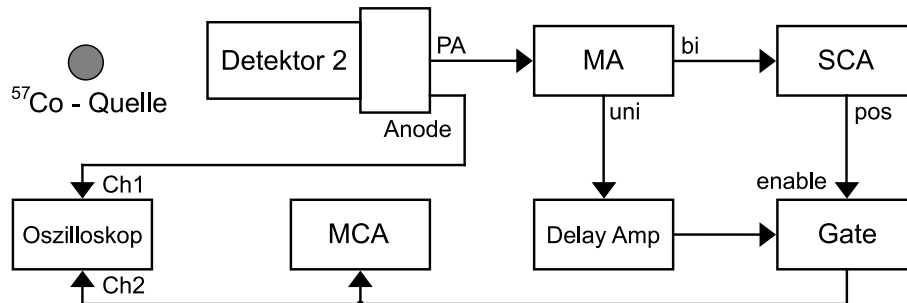


Abbildung 9: Schaltung zur Einstellung des Diskriminators auf das $14,4\text{ keV}$ -Fenster

Abbildungen 9 und 10 zeigen die Lage der Peaks im Fenster von Detektor 1, mit dem wir das 122 keV -Signal aufnehmen, einmal vor und nach unserer Hauptmessung. Hier kann man erkennen, dass sich die Lage des 122 keV -Peaks wohl leider während der Nacht ein wenig nach rechts verschoben hatte. Da aber der Peak immer noch gut zu erkennen war, hat diese leichte Verschiebung wahrscheinlich keine großen Auswirkungen auf das Ergebnis. Abbildungen 11 und 12 zeigen die Fenstereinstellung vor und nach der Messung für den anderen Detektor mit dem $14,4\text{ keV}$ -Peak. Hier ist die Lage des Peaks unverändert geblieben.

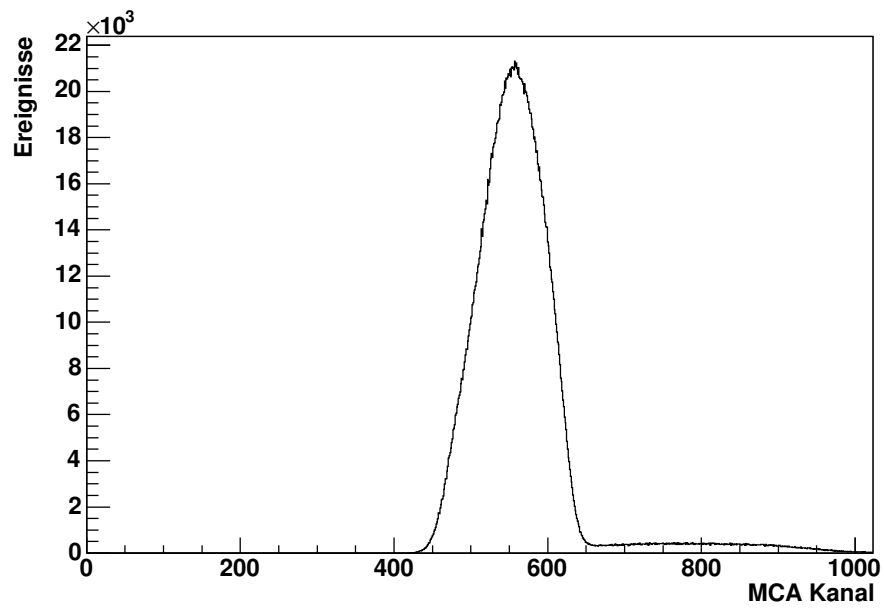


Abbildung 10: SCA-Einstellung für das Signal von Detektor 1 (vor der Messung)

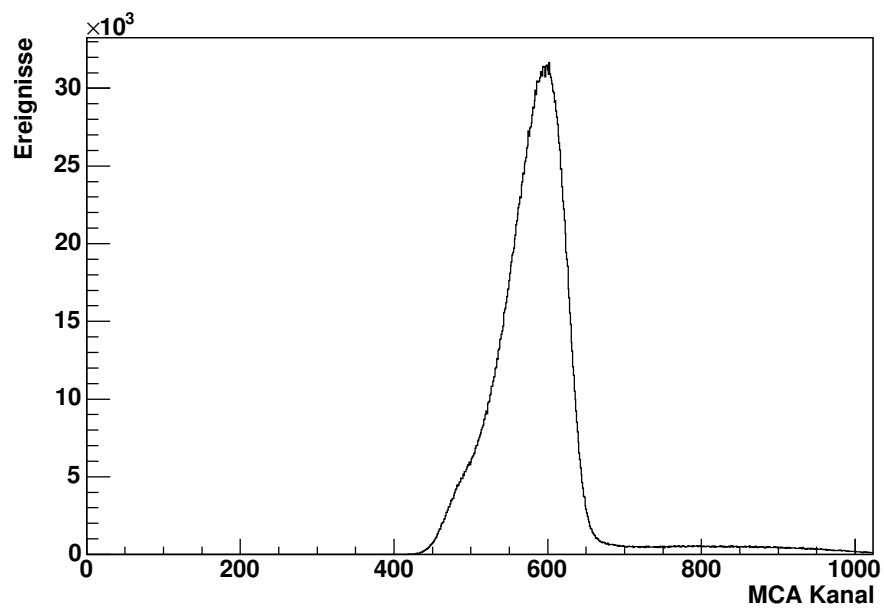


Abbildung 11: SCA-Einstellung für das Signal von Detektor 1 (nach der Messung)

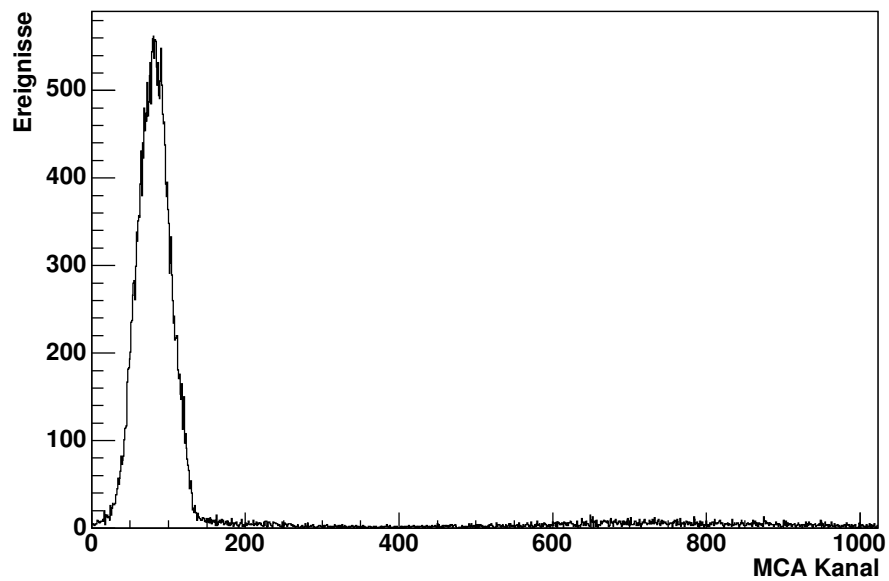


Abbildung 12: SCA-Einstellung für das Signal von Detektor 2 (vor der Messung)

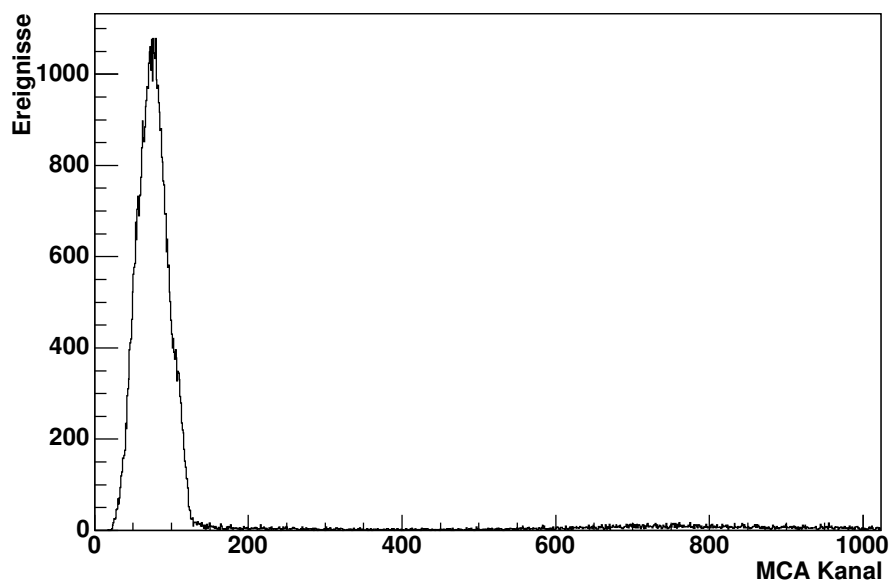


Abbildung 13: SCA-Einstellung für das Signal von Detektor 2 (nach der Messung)

Als Einstellungen zum Justieren der Fenster verwendeten wir beim MA bzw. SCA:

Signal	MA	Gain	Coarse gain	Shaping time
122 <i>keV</i>	1	5,74	100	0,5
14,4 <i>keV</i>	2	7,22	500	0,5
Signal	SCA	Upper win	Lower win	Delay
122 <i>keV</i>	1	5,60	4,20	1,00
14,4 <i>keV</i>	2	1,32	0,62	1,00

3.4. Aufnahme des Zeitspektrums

Nach Einstellung der Diskriminator-Fenster konnten wir mit der Hauptmessung beginnen. Dazu bauten die Koinzidenzschaltung in Abbildung 14 auf und verzögerten das 122 *keV*-Signal am SCA1 um 800 *ns*. Somit lag das Maximum der Zerfallskurve bei etwa 4/5 des Kanalbereichs des MCA, wodurch wir den exponentiellen Abfall über einen möglichst weiten Bereich aufzeichnen konnten. Wir ließen die Messung über Nacht laufen, die Messdauer betrug insgesamt 48653 *s* bzw. 13,5 *h*. Nach der Aufnahme des Zeitspektrums nahmen wir über 7871 *s* bzw. 2,2 *h* die zufälligen Koinzidenzen auf. Dazu stellten wir das Delay des SCA1 auf 1,0, also keine zusätzliche Verzögerung. Die Abbildungen 15 und 16 zeigen die jeweils aufgenommenen Verteilungen.

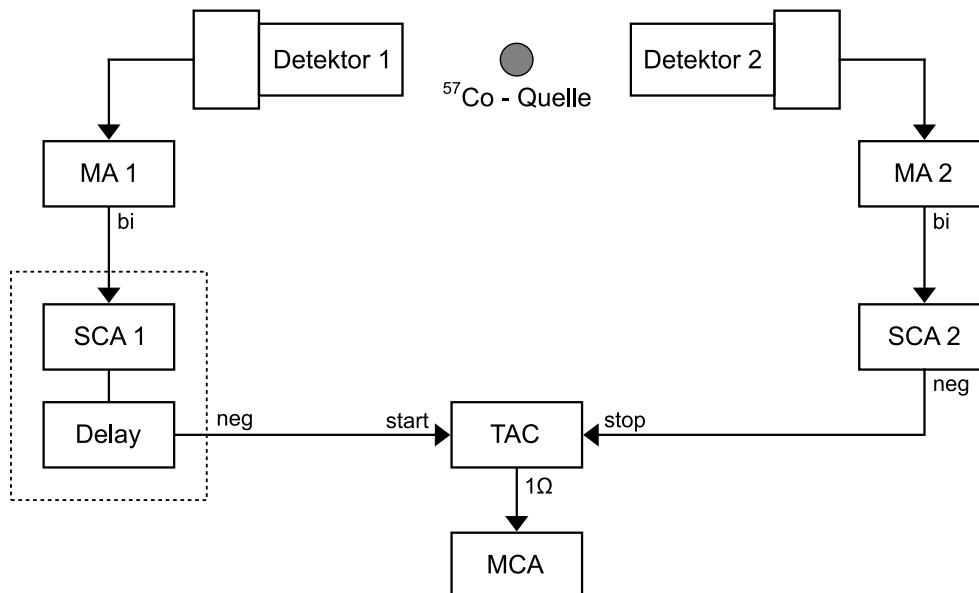


Abbildung 14: Schaltung zur Aufnahme des Zeitspektrums

Als Einstellung zur Aufnahme des Zeitspektrums verwendeten wir für die MA:

Signal	MA	Gain	Coarse gain	Shaping time
122 <i>keV</i>	1	5,74	100	0,5
14,4 <i>keV</i>	2	7,22	500	0,5

und für die SCA die Einstellungen:

Signal	SCA	Upper win	Lower win	Delay
122 <i>keV</i>	1	5,60	4,20	9,00
14,4 <i>keV</i>	2	1,32	0,62	1,00

Am TAC nahmen wir folgende Einstellungen vor:

Range	Amplitude	Gating mode
$0,1 \times 10$	ca. 6,7 Striche	Anti-Coinc.

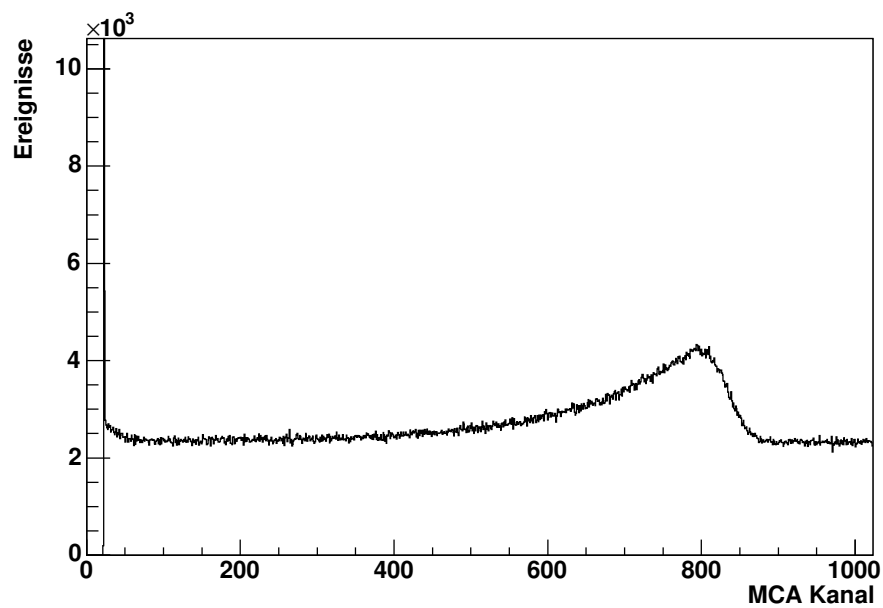


Abbildung 15: Das aufgenommene Zeitspektrum

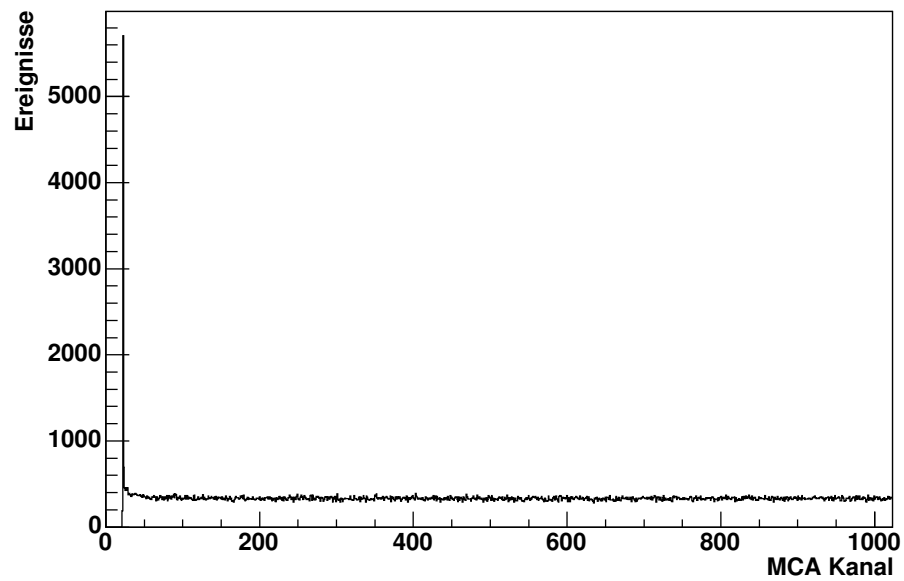


Abbildung 16: Die aufgenommenen zufälligen Koinzidenzen

4. Auswertung

4.1. Zeitkalibrierung des TAC

Um den TAC zu eichen, bauten wir verschiedene Verzögerungen in unseren Aufbau ein und beobachteten die jeweiligen Kanäle des MCA in denen dann die Signale erschienen (siehe Abschnitt 3.1). Dann trugen wir die jeweils angesprochenen Kanäle gegen die zugehörigen Verzögerungen auf (siehe Abbildung 17). Der Umrechnungsfaktor m lässt sich aus der Steigung a der Regressionsgeraden durch $m = 1/a$ berechnen, sein Fehler s_m berechnet sich aus dem Fehler s_a der Steigung durch:

$$s_m = \sqrt{\left(\frac{\partial m}{\partial a}\right)^2} s_a^2 = \left| -\frac{s_a}{a^2} \right| = \left| m \frac{s_a}{a} \right| .$$

Damit ergibt sich ein Umrechnungsfaktor von

$$m = (0,9481 \pm 0,0086) \text{ ns/channel} ,$$

das heißt, ein Kanal entspricht einem Zeitintervall von etwa $0,95 \text{ ns}$.

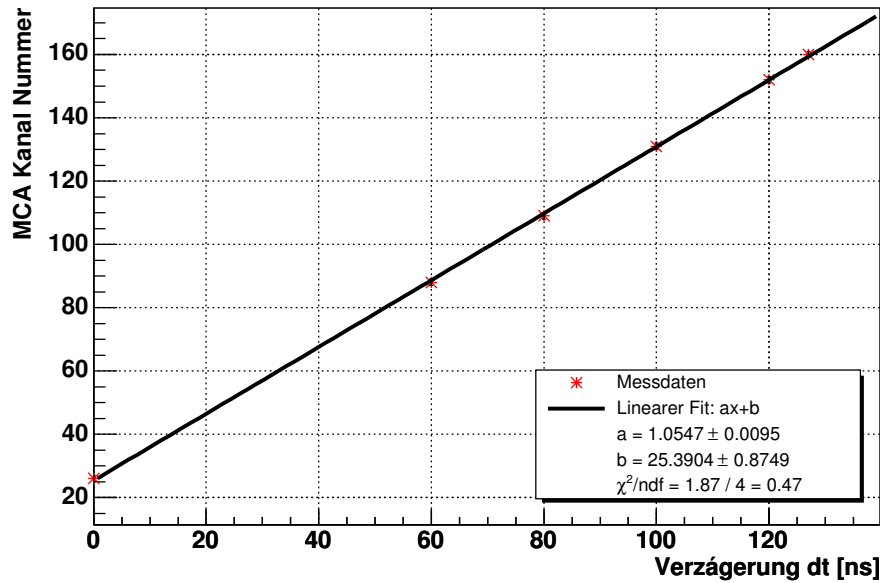


Abbildung 17: Linearer Fit zur Kalibrierung des TAC

4.2. Bestimmung der mittleren Lebensdauer und der Halbwertszeit

Zunächst zogen wir die zufälligen Koinzidenzen von dem aufgenommenen Zeitspektrum ab. Da wir die zufälligen Koinzidenzen nur über einen kürzeren Zeitraum gemessen hatten, rechneten wir die Anzahl der zufälligen Ereignisse N_u jedes Kanals auf die Messzeit des Zeitspektrums um. Wenn die Messzeit des Zeitspektrums t_s und die des Untergrunds t_u ist, dann erhält man mit den Ereignissen N_s des Zeitspektrums folgende korrigierten Ereignisse für die jeweiligen Kanäle:

$$N = N_s - \frac{t_s}{t_u} N_u \quad (6)$$

Der Fehler auf die Ereignisse der einzelnen Kanäle ist jetzt allerdings nicht mehr \sqrt{N} , sondern setzt sich aus den Fehlern der Kanäle der beiden Messreihen zusammen:

$$s_N = \sqrt{\left(\frac{\partial N}{\partial N_s}\right)^2 s_{N_s}^2 + \left(\frac{\partial N}{\partial N_u}\right)^2 s_{N_u}^2} = \sqrt{s_{N_s}^2 + \left(\frac{t_s}{t_u} s_{N_u}\right)^2} \quad (7)$$

Zur Entfernung des Untergrunds siehe auch den Quelltext in Anhang A.2.

Um nun mit dem so korrigierten Zeitspektrum die mittlere Lebensdauer und die Halbwertszeit zu bestimmen verwendeten wir die Slope-Methode (siehe Abschnitt 2.2). Dazu schätzten wir durch Spiegelung der Flanke der prompten Kurve am Maximum des Peaks ab, ab welchem Kanal die prompte Kurve keinen relevanten Beitrag zur aufgenommenen Verteilung lieferte. Bei der zu erwartenden Verteilung (Gleichung 4) ist allerdings zu beachten, dass der Exponent aufgrund der verzögerten Koinzidenz-Methode positiv ist. Als Fit-Funktion verwendeten wir also:

$$F(t) = A e^{\lambda t} + y_0 \quad (8)$$

Die Kurvenanpassung ist in Abbildung 18 zu sehen, siehe dazu auch den Quelltext in Anhang A.3.

Aus dem Fit erhielten wir für die Zerfallskonstante und deren Fehler

$$\lambda = (6,6378 \pm 0,2689) 10^{-3} \text{ channel}^{-1} .$$

Mit dem Umrechnungsfaktor m aus der Zeiteichung des TAC lässt sich daraus die mittlere Lebensdauer τ und die Halbwertszeit $T_{1/2}$ des $14,4 \text{ keV}$ -Zustands bestimmen mit:

$$\tau = \frac{m}{\lambda} \quad \text{bzw.} \quad T_{1/2} = \ln 2 \cdot \tau \quad (9)$$

Der Fehler s_τ der mittleren Lebensdauer ergibt sich dabei aus:

$$\begin{aligned} s_\tau &= \sqrt{\left(\frac{\partial \tau}{\partial m}\right)^2 s_m^2 + \left(\frac{\partial \tau}{\partial \lambda}\right)^2 s_\lambda^2} = \sqrt{\left(\frac{s_m}{\lambda}\right)^2 + \left(-\frac{m s_\lambda}{\lambda^2}\right)^2} \\ &= |\tau| \sqrt{\left(\frac{s_m}{m}\right)^2 + \left(\frac{s_\lambda}{\lambda}\right)^2} \end{aligned} \quad (10)$$

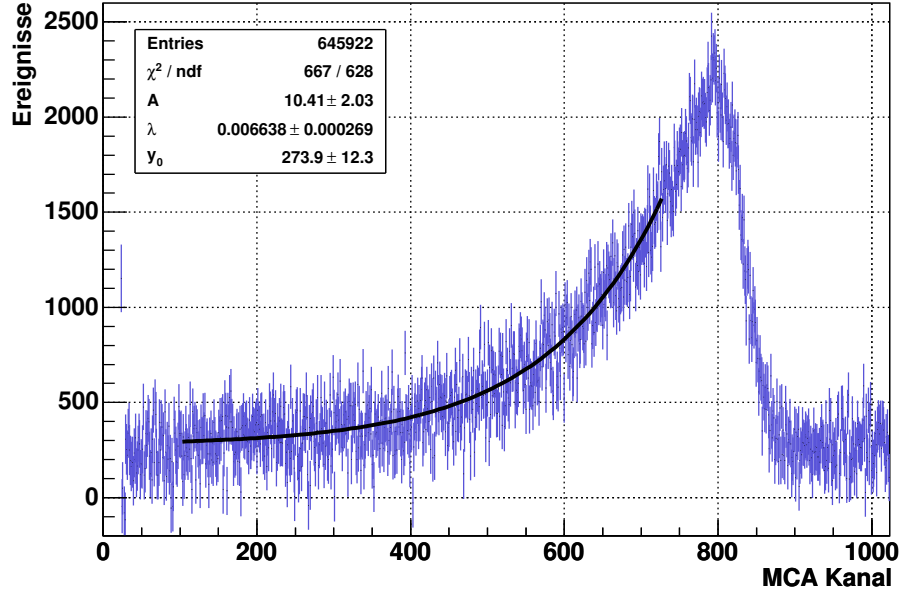


Abbildung 18: Exponentieller Fit zur Bestimmung der Zerfallskonstante λ des $14,4 \text{ keV}$ -Zustands von ^{57}Fe

und der Fehler $s_{T_{1/2}}$ der Halbwertszeit aus:

$$s_{T_{1/2}} = \sqrt{\left(\frac{\partial T_{1/2}}{\partial \tau}\right)^2 s_\tau^2} = \ln 2 \cdot s_\tau . \quad (11)$$

Damit erhielten wir als mittlere Lebensdauer

$$\tau = (142,8 \pm 5,9) \text{ ns}$$

und für die Halbwertszeit:

$$T_{1/2} = (99,0 \pm 4,1) \text{ ns} .$$

5. Zusammenfassung

Als mittlere Lebensdauer für den $14,4\text{ keV}$ -Zustand von ^{57}Fe ermittelten wir:

$$\tau = (142,8 \pm 5,9) \text{ ns}$$

und für die Halbwertszeit:

$$T_{1/2} = (99,0 \pm 4,1) \text{ ns} .$$

Unser Fehler von immerhin 4,15% ist allerdings recht hoch, was auf die leider etwas kürzere Messung der zufälligen Koinzidenzen zurück zu führen ist.

Der Wert für die mittlere Lebensdauer weicht um $(2,6 \pm 5,9) \text{ ns}$ von dem von Clark¹ bestimmten Wert von $\tau_{\text{Clark}} = (140,2 \pm 0,3) \text{ ns}$ ab und liegt also im $0,44\sigma$ -Bereich der Fehler.

Beim Vergleich mit dem Literaturwert² für die Halbwertszeit von $T_{1/2, \text{Lit}} = (98,3 \pm 0,3) \text{ ns}$ erhält man eine Differenz von $(0,7 \pm 4,1) \text{ ns}$. Die Werte liegen in diesem Fall also im $0,17\sigma$ -Bereich ihrer Fehler.

Insgesamt gibt es also eine gute Übereinstimmung der von uns bestimmten Werte mit dem Literaturwert bzw. dem Wert von Clark.

¹Aus *R. E. BELL: Coincidence Techniques and the Measurement of Short Mean Lives*

²Quelle: National Nuclear Data Center, <http://www.nndc.bnl.gov/>

A. Quelltexte ³

Wir verwendeten zur Auswertung die Programmiersprache Python mit dem Datenanalyse Framework ROOT.

A.1. Kalibrierung des TAC (tac_kalib.py)

```
1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from array import array
5  from ROOT import gROOT, TCanvas, TGraphErrors, TF1, TLegend
6
7  gROOT.SetStyle("Plain")
8
9  # -----
10 # Kalibrierung des TAC
11 # -----
12
13 # Messdaten (Delay, Channel)
14 data = [(0, 26), (60, 88), (80, 109), (100,131), (120,152), (127, 160)]
15
16 # Delay [ns]
17 dt = array('d', [z[0] for z in data])
18 sdt = array('d', [0]*len(dt))
19
20 # Channel Nr
21 ch = array('d', [z[1] for z in data])
22 sch = array('d', [1]*len(ch))
23
24 # Erzeuge Graphen
25 g = TGraphErrors(len(data), dt, ch, sdt, sch)
26 g.SetTitle(';Verzögerung dt [ns];MCA Kanal Nummer')
27 g.GetHistogram().SetTitleOffset(1.1, 'Y')
28 g.SetMarkerColor(2)
29 g.SetMarkerStyle(3)
30
31 # Fitte die Messdaten linear
32 f = TF1('f', '[0]*x + [1]',0,140)
33 g.Fit(f, 'RQ')
34 a, b = f.GetParameter(0), f.GetParameter(1)
35 sa, sb = f.GetParError(0), f.GetParError(1)
36 chisq, ndf = f.GetChisquare(), f.GetNDF()
37 rchisq = chisq/ndf
38
```

³Siehe <http://www.physik.uni-freiburg.de/~kolja/fp1/khwz/>.

```

39 # Umrechnungsfaktor m von Kanal-Nr. in Delay [ns/ch]
40 m = 1./a
41 sm = m * sa/a
42
43 # Gebe Ergebnis aus
44 print 'Linearer Fit: a*x + b'
45 print 'a = %f +- %f' % (a, sa)
46 print 'b = %f +- %f' % (b, sb)
47 print 'rchisq = %.2f/%d = %.2f' % (chisq, ndf, rchisq)
48 print 'm = %f +- %f' % (m, sm)
49
50 # Erzeuge Legende
51 l = TLegend(0.55, 0.14, 0.88, 0.34)
52 l.SetFillColor(0)
53 l.AddEntry(g, 'Messdaten', 'p')
54 l.AddEntry(f, 'Linearer Fit: ax+b', 'l')
55 l.AddEntry(f, 'a = %.4f #pm %.4f' % (a, sa), '')
56 l.AddEntry(f, 'b = %.4f #pm %.4f' % (b, sb), '')
57 l.AddEntry(f, '#chi^2/ndf = %.2f / %d = %.2f' % (chisq, ndf, rchisq), '')
58
59 # Zeichne Ergebnis
60 c = TCanvas('c', 'TAC-Kalibrierung')
61 c.SetGrid()
62 g.Draw('AP')
63 l.Draw()
64 c.Update()

```

Programmausgabe

```

1 Linearer Fit: a*x + b
2 a = 1.054738 +- 0.009534
3 b = 25.390410 +- 0.874947
4 rchisq = 1.87/4 = 0.47
5 m = 0.948103 +- 0.008570

```

A.2. Erzeugung der Histogramme aus den Messdaten (messung.py)

```

1 #!/usr/bin/python
2 # -*- coding: iso-8859-1 -*-
3
4 from math import sqrt
5 from ROOT import TH1F
6
7 # -----
8 # Klassen zur Handhabung der Messungen
9 # -----
10
11 # Klasse zum Erstellen von Histogrammen aus den Messdaten

```

```

12 class Messung:
13     def __init__(self, name):
14         self.name = name
15         i, header = 1, True
16         h = TH1F(name, name, 1024, 0, 1023)
17         h.SetTitle(';MCA Kanal;Ereignisse')
18         for line in open('messungen/'+name, 'r'):
19             if not header:
20                 tokens = line.split(',')
21                 h.SetBinContent(i, int(tokens[1]))
22                 i += 1
23             elif line.strip()[0:3] == 'Chn':
24                 header = False
25             else:
26                 tokens = line.split()
27                 if len(tokens) > 1 and tokens[-2] == 'Time:':
28                     self.time = int(tokens[-1])
29         h.SetEntries(h.GetSum())
30         self.histo = h
31         self.bins = h.GetNbinsX()
32         self.count = h.GetSum()
33
34 # Erweiterung der Klasse zum Entfernen des Untergrunds
35 class MessungZeitspek(Messung):
36     def entferne_untergrund(self, m):
37         assert self.bins == m.bins
38         for i in range(1, self.bins+1):
39
40             # Lese Bin der Messung und des Untergrunds aus
41             s = self.histo.GetBinContent(i)
42             ss = self.histo.GetBinError(i)
43
44             # Rechne den Untergrund auf die Zeit der Hauptmessung um
45             u = m.histo.GetBinContent(i) * float(self.time) / float(m.time)
46             su = m.histo.GetBinError(i) * float(self.time) / float(m.time)
47
48             # Ziehe die Untergrundereignisse ab und berücksichtige die Fehler
49             v = s - u
50             sv = sqrt(ss**2 + su**2)
51
52             # Schreibe das Ergebnis in das aktuelle Bin
53             self.histo.SetBinContent(i, v)
54             self.histo.SetBinError(i, sv)
55
56             # Korrigiere die Zahl der Gesamtereignisse
57             self.count = self.histo.GetSum()
58             self.histo.SetEntries(self.count)

```

A.3. Bestimmung der Lebensdauer und der Halbwertszeit (lebensdauer.py)

```
1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from messung import *
5  from array import array
6  from math import sqrt, log
7  from ROOT import gROOT, TCanvas, TH1F, TF1, TPaveStats
8
9  gROOT.SetStyle("Plain")
10
11  # -----
12  # Bestimmung der Lebensdauer des 14,4keV Zustands
13  # -----
14
15  # Umrechnungsfaktor von der Kanalnummer in Zeit [ns] (aus tac_kalib.py)
16  m, sm = (0.94810250222639858, 0.0085703351098888221)
17
18  # Lese Untergrund ein
19  mug = Messung('nullspek.asc')
20  hug = mug.histo
21
22  # Lese Zeitspektrum ein und entferne Untergrund
23  mspek = MessungZeitspek('zeitspek.asc')
24  mspek.entferne_untergrund(mug)
25  hspek = mspek.histo
26
27  # Fuehre Fit durch
28  f = TF1('f', '[0] * exp([1]*x) + [2]', 100, 730)
29  f.SetParameters(array('d', [10, 0.001, 150]))
30  f.SetParName(0, 'A'); f.SetParName(1, '#lambda'); f.SetParName(2, 'y_{0}')
31  hspek.Fit(f, 'RQ0')
32  A, l, y0 = f.GetParameter(0), f.GetParameter(1), f.GetParameter(2)
33  sA, sl, sy0 = f.GetParError(0), f.GetParError(1), f.GetParError(2)
34  chisq = f.GetChisquare()
35  ndf = f.GetNDF()
36  rchisq = chisq/ndf
37
38  print 'Exponentieller Fit: A * exp(lambda * t) + y0'
39  print 'A = %g +- %g' % (A,sA)
40  print 'lambda = %g +- %g' % (l,sl)
41  print 'y0 = %g +- %g' % (y0,sy0)
42  print 'chisq/ndf = %.2f / %d = %.2f' % (chisq, ndf, rchisq)
43
44
45  # Berechne mittlere Lebensdauer [ns]
46  tau = m / l
47  stau = tau * sqrt( (sm/m)**2 + (sl/l)**2 )
```



```

48
49 # Berechne Halbwertszeit [ns]
50 T = log(2) * tau
51 sT = log(2) * stau
52
53 print 'Mittlere Lebensdauer tau = %f +- %f' % (tau, stau)
54 print 'Halbwertszeit T12 = %f +- %f' % (T, sT)
55
56
57 # Zeiche Untergrund Histogramm
58 cug = TCanvas('cug', 'Untergrund')
59 cug.SetGrid()
60 hug.SetLineColor(9)
61 hug.Draw()
62 cug.Update()
63
64 # Zeichne Lebensdauer Histogramm mit Fit
65 c = TCanvas('c', 'Bestimmung der Lebensdauer')
66 c.SetGrid()
67 hspek.SetLineColor(9)
68 hspek.SetTitleOffset(1.3, 'Y')
69 hspek.SetMinimum(-200)
70 hspek.SetMaximum(2600)
71 pave = hspek.GetListOfFunctions().FindObject('stats')
72 hspek.Draw()
73 f.Draw('same')
74 c.Update()
75
76 # Berechne Abweichungen
77 Tlit, sTlit = 98.3, 0.3
78 dT = T-Tlit
79 sdT = sqrt(sT**2 + sTlit**2)
80 print 'Differenz von T1/2 zum Literaturwert: %g +- %g' % (dT, sdT)
81
82 tau_clark, stau_clark = 140.2, 0.3
83 dtau = tau-tau_clark
84 sdttau = sqrt(stau**2 + stau_clark**2)
85 print 'Differenz von tau zum Wert von Clark: %g +- %g' % (dtau, sdttau)

```

	Programmausgabe
1	Exponentieller Fit: $A * \exp(\lambda * t) + y_0$
2	$A = 10.4115 \pm 2.02704$
3	$\lambda = 0.00663776 \pm 0.000268931$
4	$y_0 = 273.889 \pm 12.3055$
5	$\chi^2/\text{ndf} = 667.03 / 628 = 1.06$
6	Mittlere Lebensdauer $\tau = 142.834813 \pm 5.929284$
7	Halbwertszeit $T_{12} = 99.005548 \pm 4.109867$
8	Differenz von $T_{1/2}$ zum Literaturwert: 0.705548 ± 4.1208
9	Differenz von τ zum Wert von Clark: 2.63481 ± 5.93687

A.4. Anzeigen der Histogramme (showhistos.py)

```
1  #!/usr/bin/python
2  # -*- coding: iso-8859-1 -*-
3
4  from os import listdir
5  from messung import Messung
6  from ROOT import gROOT, TCanvas, TH1F, TGaxis
7
8  gROOT.SetStyle("Plain")
9  TGaxis().SetMaxDigits(4)
10
11  m, c = [], []
12  name = filter(lambda z: z[-4:] == '.asc', listdir('messungen'))
13
14  for n in name:
15      mi = Messung(n)
16      ci = TCanvas('c_'+n, n)
17      mi.histo.Draw()
18      ci.Update()
19
20      m += [mi]; c += [ci]
```