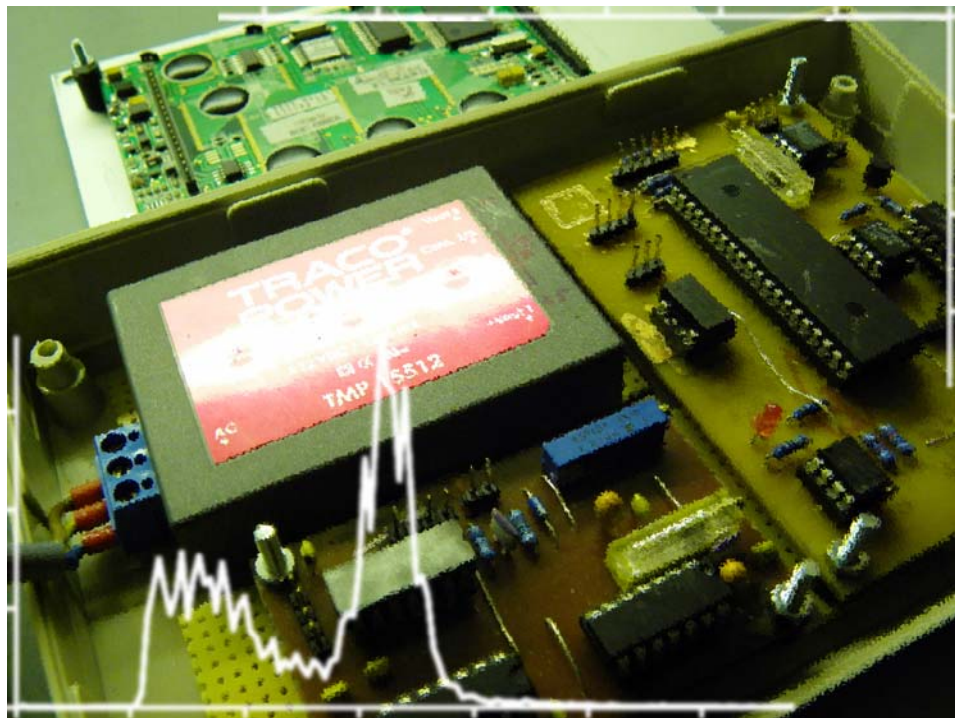


# Selbstgebauter Vielkanalanalysator zur Gammaskopie



**Wettbewerb „Jugend Forscht“ 2011**

**Lucas Jürgens (15 Jahre)**

**Arbeitsgemeinschaft "Jugend Forscht"  
des Christian-Gymnasiums Hermannsburg  
Leitung: StD Thomas Biedermann**

# Inhaltsverzeichnis

1	Allgemein .....	1
1.1	Was ist ein Vielkanalanalysator? .....	1
1.2	Szintillationszählrohr .....	2
2	Elektronik .....	2
2.1	Messelektronik .....	3
2.2	Steuerelektronik .....	4
2.2	Steuerelektronik .....	5
3	Software .....	6
3.1	Programmstruktur .....	6
3.2	Messvorgang .....	7
3.3	DA-Wandler .....	7
3.4	Display .....	8
4	Praxis .....	9
4.1	Messungen .....	9
4.1.1	Caesium 137 .....	9
4.1.2	Eichung .....	9
4.1.3	Einfluss der SZR-Spannung auf das Spektrum .....	10
4.1.4	Natrium 22 .....	10
4.1.5	Cobalt 60 .....	11
4.1.6	Umgebungsstrahlung .....	11
4.1.6	Umgebungsstrahlung .....	12
4.2	Kosten .....	12
5	Quellen, Literatur, Werkzeuge .....	13
6	Danksagung .....	13

# 1 Allgemein

Das Ziel meines Projektes ist es, einen Vielkanalanalysator zu bauen, der die Impulse eines Szintillationszählrohres (SZR) erfasst und ein Energiespektrum des gemessenen radioaktiven Präparates erstellt. Das Gerät sollte dabei möglichst benutzerfreundlich und günstig sein, aber trotzdem gute Messergebnisse liefern. Das Projekt baut dabei auf meinem Projekt vom letzten Jahr „Selbstgebautes portables digitales PIC-gestütztes Oszilloskop“ auf.

## 1.1 Was ist ein Vielkanalanalysator?

Ein Vielkanalanalysator (VKA) sortiert regellos eintreffende Impulse nach deren Höhe in verschiedene Kanäle, wobei die Impulse gleicher Amplitude aufsummiert werden. Üblich sind VKAs mit 64 bis 256 Kanälen und einer Zähltiefe von bis zu 64.000 Impulsen pro Kanal.

Da in meinem Fall zur Gammaskopie die Impulse von einem SZR stammen, entspricht die Häufigkeitsverteilung dann dem Spektrum der untersuchten Strahlung.

Weil die Energie der Gammastrahlung bei radioaktiven Zerfallsprozessen je nach Isotop unterschiedlich ist, kann man so unbekannte radioaktive Stoffe, auch Stoffgemische, eindeutig identifizieren.

Um die Energie einem Kanal zuzuordnen, muss man die Messung mit einem Stoff bekannter Energie (z.B. Cs137) eichen.

Der von mir gebaute VKA verfügt momentan über 240 Kanäle mit einer Zähltiefe von 65536 Impulsen pro Kanal. Die Messungen können gespeichert und abgerufen werden. Alle wichtigen Messparameter sowie das Spektrum werden während des Messvorgangs auf dem Display ausgegeben. Über dieses können per Touchscreen Einstellungen wie die Messzeit, die Betriebsspannung für die SZR, Anzahl der Kanäle oder Darstellungseinstellungen vorgenommen werden. Die Messungen können später über eine RS-232 Schnittstelle auf den PC zur weiteren Auswertung übertragen werden.

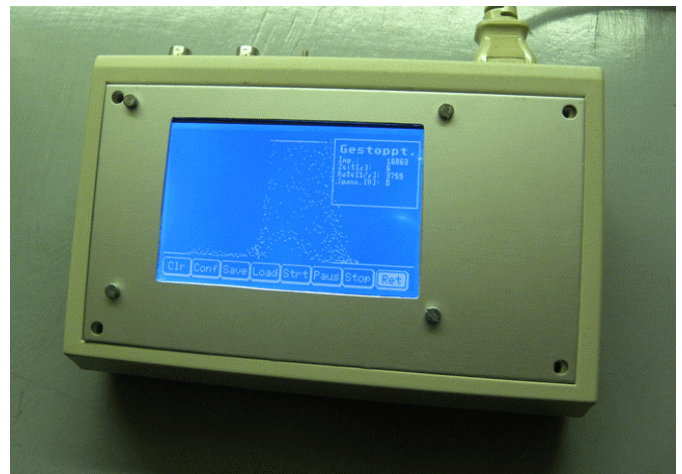


Abb. 1: VKA im Gehäuse

## 1.2 Szintillationszählrohr

Abbildung 2 zeigt den prinzipiellen Aufbau eines Szintillationszählrohrs. Der Szintillationskristall im Kopf des Rohres besteht aus Natriumjodid, in dem die Energie eines eintreffenden Gammaquants absorbiert wird. Die dadurch entstandenen Elektronen regen die Atome zur Emission von Licht an. Je höher die Energie des Gammaquants, desto mehr Photonen werden freigesetzt. Diese treffen dann auf eine Photokathode und lösen jeweils ein Elektron heraus, welches anschließend durch ein elektrisches Feld auf eine Dynode beschleunigt wird, welche weitere Elektronen emittiert. Auch diese werden weiter beschleunigt und treffen auf eine weitere Dynode. Durch mehrere solcher Schritte wird am Ende auf der Anode ein Stromimpuls erzeugt, der proportional zur Anzahl der auf der Photokathode aufgetroffenen Photonen und somit der Energie des Gammaquants ist und ohne weiteres gemessen werden kann.

Die Dynoden werden dabei mit einer von der ersten zur letzten Dynode steigenden Spannung versorgt. Je höher die Spannung, desto höher ist die Vermehrungsrate pro Dynode und umso größer die resultierende Verstärkung.

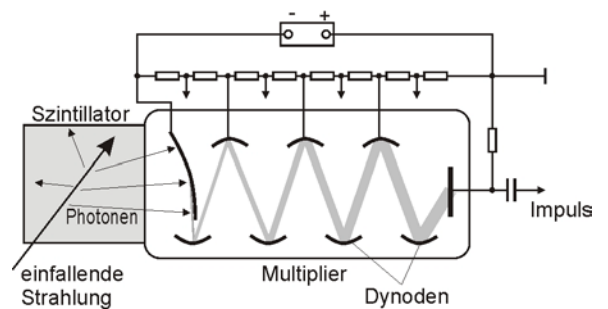


Abb. 2: Aufbau eines Szintillationszählrohrs mit Szintillationskristall (links), Multiplier und die zugehörige Beschaltung für die Dynoden

## 2 Elektronik

Abbildung 3 zeigt die gesamte Elektronik des VKA. Oben sieht man die Rückseite des Touch-Displays eDIP240-7 von „Elektronik Assembly“ mit 240 x 128 Pixeln. Darunter (blau eingerahmt) habe ich ein Schaltnetzteil eingebaut, welches +/- 15V für die Analogelektronik und 5V für den Mikrocontroller liefert. Darunter befindet sich die Analogelektronik, die die Impulse der SZR verarbeitet. Rechts daneben ist das PIC-Steuerboard. Wie man in Abb.4 sieht, verfügt der VKA über ein Kaltgerätekabelanschluss, über den er mit 230V betrieben werden kann. Über die SUB-D Buchse rechts daneben kann der Mikrocontroller programmiert und die Messungen übertragen werden.

Der linke BNC-Anschluss ist der Impulseingang, der rechte liefert die Steuerspannung (0-10V) für das zum Betrieb des Zählrohrs erforderliche Hochspannungsnetzteil.

Die Platinen habe ich mit Eagle der Version 5.10.0 layoutet und per Tonerverfahren auf eine Kupferplatine aufgebracht, geätzt und gebohrt.

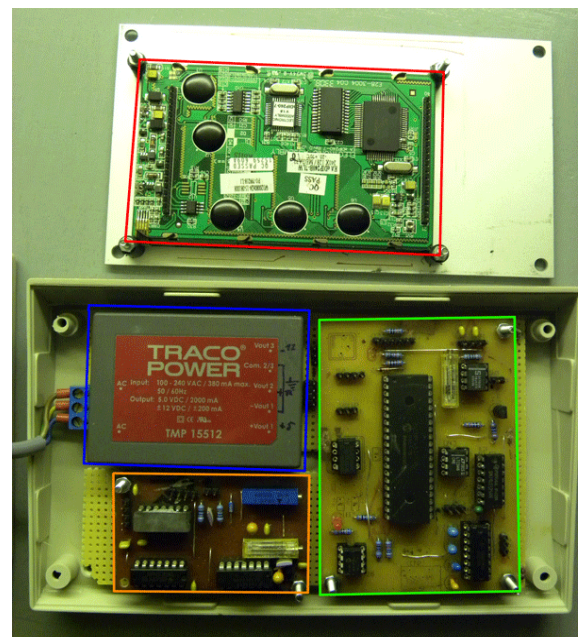


Abb. 3: VKA: Elektronik in Gehäuse



Abb. 4: VKA: Von hinten

## 2.1 Messelektronik

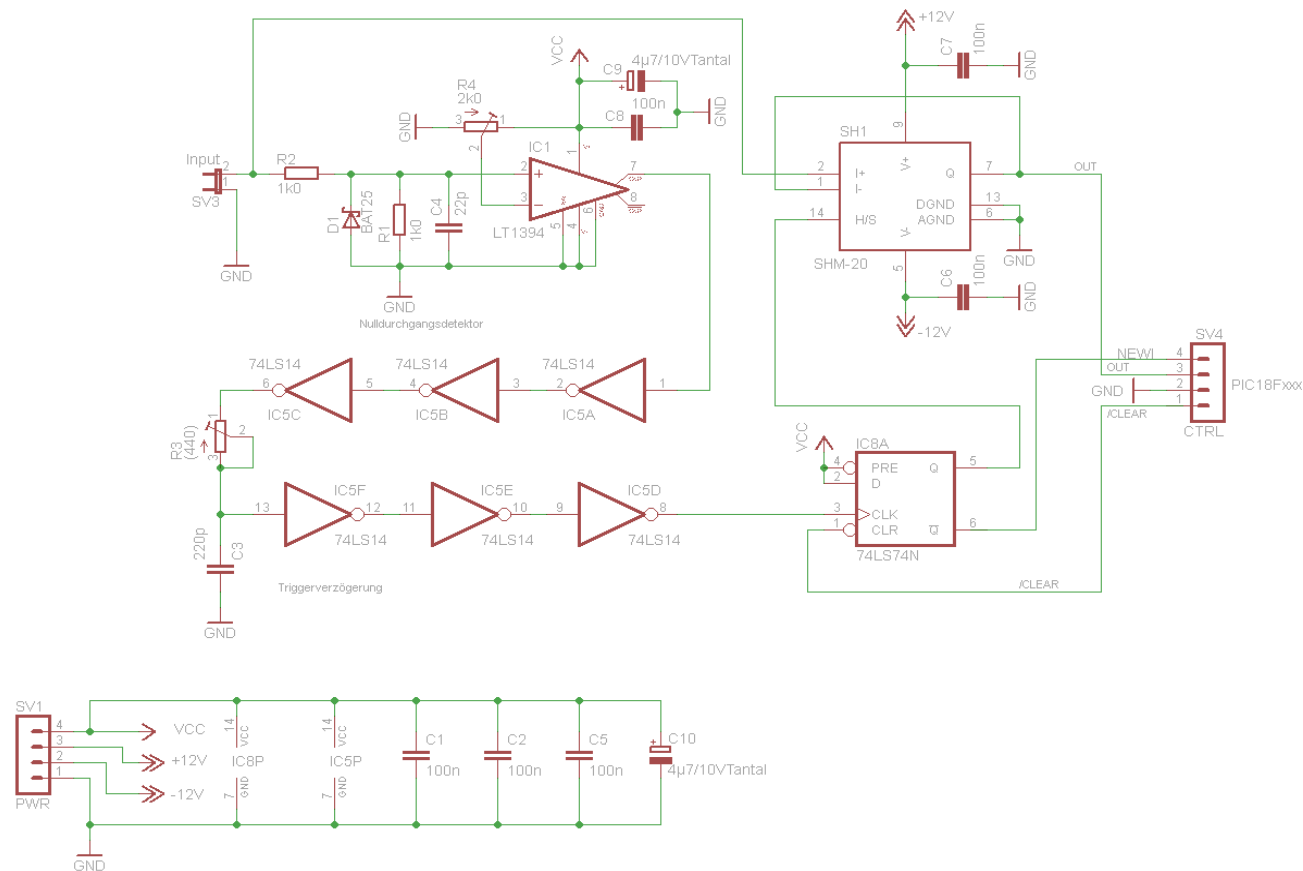


Abb. 5: Schaltplan der Messelektronik

SV1: Steckverbinder Stromversorgung  
 SV3: Steckverbinder Impulseingang  
 SV4: Steckverbinder PIC

IC1: Nulldurchgangsdetektor  
 SH1: „Sample and Hold“ Baustein  
 IC8: Flipflop  
 IC5: Inverter

Ein Impuls des SZR hat eine Dauer von ca. 200ns Sekunden, die maximale Impulshöhe wird nach ca. 100 ns erreicht. Da es für den PIC unmöglich wäre mit seinem internen Analog-Digital Wandler (ADC) so kurze Impulse zu erkennen und auszuwerten, musste ich eine schnelle Analogspeicherschaltung entwickeln. Deren Aufgabe besteht darin, den Impuls zu erkennen und die Spitze des Impulses festzuhalten, sodass der PIC sie auswerten kann.

Die Messelektronik besteht aus 3 wesentlichen Komponenten; einem Nulldurchgangsdetektor, einem „Sample and Hold“-Baustein (SH) und einer Triggervverzögerung.

Zur Triggervverzögerung habe ich einen Hex-Inverter zweckentfremdet und die einzelnen Inverter in Reihe geschaltet, um die Latenzzeit der einzelnen Gatter auszunutzen, zusätzlich erlaubt ein einstellbares RC-Glied eine genaue Justierung dieser Verzögerung.

Trifft am Eingang SV3 ein Impuls ein, löst der Nulldurchgangsdetektor (IC1) mit einer Verzögerung von max. 7 ns aus und legt den Ausgang Pin 7 auf „High“, wodurch die Triggervverzögerung gestartet wird. Nach ca. 100 ns (diese Zeit ist mit P3 einstellbar) wird dann der Flipflop auf „High“ gesetzt. Der Flipflop setzt dann über den Ausgang „Q“ den SH in den „Hold“-Zustand und speichert somit das zu dem Zeitpunkt am Eingang (2)

anliegende Signal. Erst wenn der PIC über die „clear“-Leitung den Flipflop zurücksetzt, wird der SH zurückgesetzt und ein neuer Impuls kann eintreffen und den Flipflop erneut auslösen. Über den FlipFlop-Ausgang Pin 6 weiß der PIC, dass ein neuer Impuls bereitsteht, über den Ausgang am Pin 7 des SH erhält der PIC die Spannung des letzten Impulses, den er über den internen ADC auswertet.

Mit dem Potentiometer R4 kann der Nullpunkt und mit Potentiometer R3 die Triggervverzögerung eingestellt werden.

Auf den Abbildungen sieht man Oszilloskop-Aufnahmen von dem Verhalten der Schaltung bei Impulsen verschiedener Höhe. Der SH wird hier zum Test automatisch zurückgesetzt.

Bei einer Impulshöhe von 2200 mV und 4700 mV wird der Impuls richtig getriggert und die richtige Höhe wird gehalten. Bei einer Spannung von 5800 mV allerdings nimmt der SH einen falschen Wert an. Das liegt daran, dass wegen der begrenzten Anstiegsgeschwindigkeit mit steigender Höhe das Impulsmaximum später erreicht wird und der Abstand von der Nullpunktüberschreitung bis zur Spitze größer ist, aber die Schaltung zum fest eingestellten Zeitpunkt, damit aber vor Erreichen des Impulsmaximums, triggert.

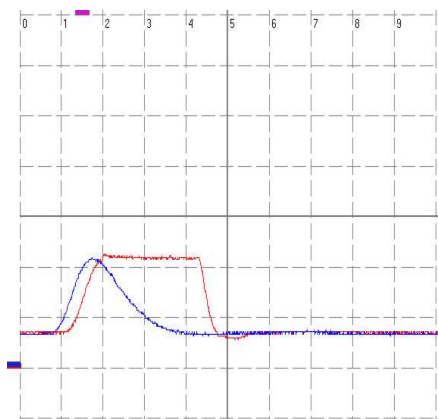


Abb. 6: X: 100ns pro Einheit, Y: 1 V pro Einheit, Blau: Impuls 2200 mV, Rot: "Sample and Hold"-Ausgang

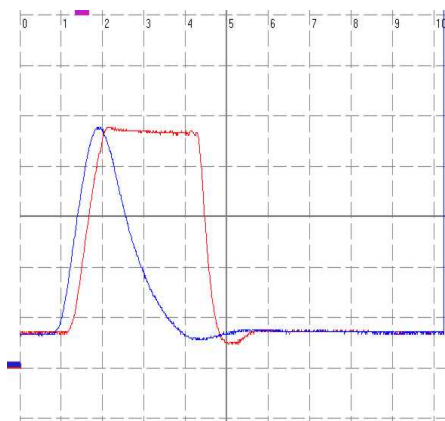


Abb. 7: Impuls 4700 mV

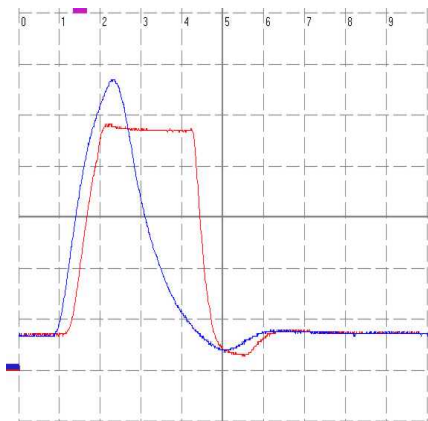


Abb. 8: Impuls 5800 mV

## 2.2 Steuerelektronik

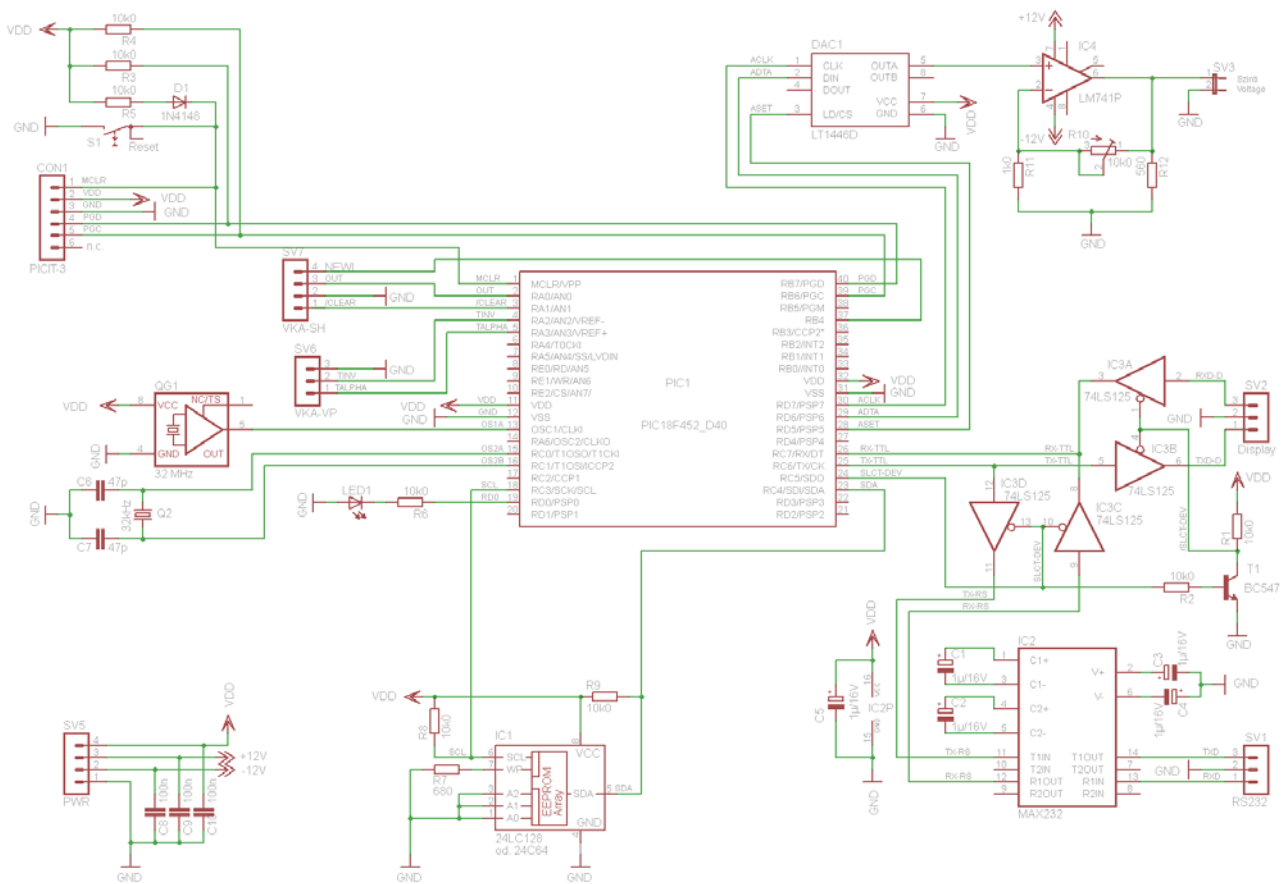


Abb. 9: Schaltplan der PIC-Platine

Con 1: Anschluss „PicKIT3“

SV1: Steckverbinder PC, RS232

SV2: Steckverbinder Display

SV3: Steckverbinder Hochspannungsnetzteil

SV5: Steckverbinder Stromversorgung

SV6: Steckverbinder Verstärkerelektronik

SV7: Steckverbinder Messelektronik

DAC1: DA-Wandler

IC4: Operationverstärker, Steuerspannung

IC2: Pegelwandler TTL → PC

IC1: I<sup>2</sup>C-Speicherbaustein

IC3: Umschalter PC ↔ Display

QG1: 40MHz Oszillator

Gesteuert wird der VKA vom einem PIC-Mikrocontroller von Microchip. Ich verwende wieder den PIC18F452 in der DIP-Bauform. Hinter dem Digital-Analog-Wandler (DAC) ist ein Operationsverstärker geschaltet, welcher die Spannung des DAC (0-4V) verstärkt, sodass der PIC das Hochspannungsnetzteil voll aussteuern kann (0-10V). Die erforderliche Verstärkung kann mit dem Potentiometer R10 eingestellt werden. Mit dem Port RC5 kann der PIC auswählen ob mit Display oder RS232-Schnittstelle kommuniziert werden soll. Darunter (IC2) befindet sich der Pegelwandler MAX232, der die TTL-Signale (0-5V) des PICs in den vom PC verwendeten Pegel des COM-Ports (-/+12V) umwandelt. IC1 ist ein 128KBit EEPROM-Speicherbaustein, der um 2 Leitungen mit dem PIC verbunden ist und später die Messungen speichern soll.

### 3 Software

Damit der PIC-Mikrocontroller Signale verarbeitet, muss man ihn vorher mit einem Programm versehen. Ich verwende dazu die Software MPLAB (Version 8.36) und dem MC18-Compiler von Microchip (Version 3.36). Der Compiler wandelt den in der Programmiersprache „C“ geschriebenen Code in Maschinencode für den PIC um, welcher dann mit dem PC über das USB-Programmierinterface „PIC-KIT 3“ der Firma Microchip in den Mikrocontroller geschrieben werden kann. Dieses Interface besitzt auch einen Debugmodus, über den die Variablen des Mikrocontrollers beobachtet und Breakpoints gesetzt werden können. Das erleichtert die Fehlersuche enorm. Der Compiler unterstützt modulare Programmierung, sodass die einzelnen Programmteile in separaten Files erstellt und getestet werden können.

#### 3.1 Programmstruktur

In Abb. 10 ist die Grundstruktur des Hauptprogramms in Form eines Strukturgramms dargestellt, das direkt nach dem Einschalten ausgeführt wird.

Zuerst initialisiert sich der PIC, d.h. es werden Variablen definiert, das Display gestartet und Ein- und Ausgänge festgelegt. Wenn nun der Messvorgang aktiv ist (Sample=1) und ein neuer Impuls beim Flipflop registriert wurde (newImp=1), wird „doSample()“ ausgeführt und der Impuls ausgewertet. Wenn jetzt durch die Interrupt-Funktion (s. weiter unten) die Event-Variable gesetzt wurde und somit einen anderen Wert als „0“ hat, wird diese zurückgesetzt und die zugehörige Funktion (1, 2 oder 3) ausgeführt.

Die Interrupt-Routine läuft parallel zur Main-Schleife. Ein Interrupt tritt ein, wenn entweder ein neues Zeichen vom Display an der USART-Schnittstelle empfangen wurde oder der Timer abgelaufen ist (Abb. 11). Wenn ein neues Byte vom Display empfangen wurde, wird dieses dem Empfangsbuffer angehängt. Wenn der Befehl vollständig ist, wird er verarbeitet und ausgewertet und ein entsprechendes Event wird gesetzt. In jedem Fall wird nach einem USART-Ereignis das USART-Flag zurückgesetzt. Ist der Timer übergelaufen (Timer.Flag=1), wird ein Counter weitergezählt. Da der Timer-Interrupt regelmäßig ausgeführt wird, kann mit der nachfolgenden Bedingung das Zeitintervall festgelegt werden. In diesem Beispiel wird das Event gesetzt, wenn der Counter den Wert 150 überschreitet. Auch hier werden zum Schluss der Timer und das Timer-Flag zurückgesetzt.

Diese Event-Steuerung ist nötig, weil in der Interrupt-Routine

Grundstruktur

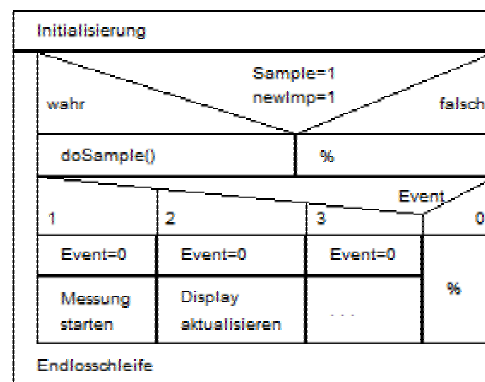


Abb. 10: Strukturgramm „main()“

Interrupt

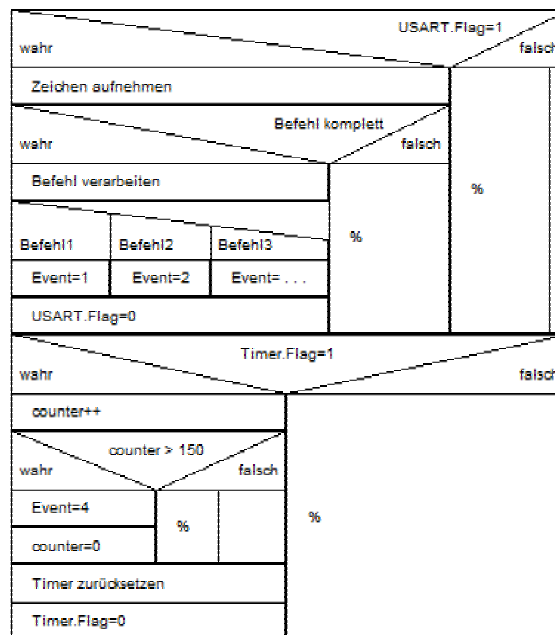


Abb. 11: Strukturgramm Interrupt-Routine



keine zeitaufwändigen Funktionen, wie zum Beispiel Displayausgaben oder Benutzereingaben ausgeführt werden dürfen, da sich sonst zeitkritische Aufgaben, wie das Empfangen von Befehlen, verzögern würden.

### 3.2 Messvorgang

Da der MC18-Compiler großen Arrays normal deklariert nicht unterstützt, müssen sie mit „#pragma udata“ umgeben werden.

Selbst dann darf das Array nur 256 Byte groß sein. Deshalb verwende ich 2 Integer-Arrays mit einer Größe von 128, um 240 Kanäle aufzeichnen zu können. Damit ich diese ähnlich wie ein einziges Array behandeln kann, habe ich die Funktion „Incspec(int n)“ und „spec(int n)“ geschrieben, die dies ermöglichen.

```
#pragma udata spec1u
    int spec2[128];
#pragma udata

#pragma udata spec2u
    int spec1[128];
#pragma udata
```

Abb. 0: C-Code:  
Deklaration der Mess-  
Variablen

Abb. 13 zeigt die Funktion „doSample()“ die ausgeführt wird, wenn ein neuer Impuls eingetroffen ist. Zu Debugzwecken wird zuerst die LED eingeschaltet, die am Ende wieder ausgeschaltet wird. Danach wird dem Integer „temp“ der um Faktor vier herunter-skalierte ADC-Wert zugewiesen. Anschließend werden die „spec“-Variablen, die die Daten des Spektrums enthalten, an der Stelle „temp“ um einen Zähler erhöht. Die nächste Bedingung trifft zu, wenn das Spektrum zur korrekten Darstellung auf dem Display skaliert werden muss. „Shalfts“ enthält den Teilfaktor. Wenn die durch „Shalfts“ geteilte Impulsanzahl für den aktuellen Kanal also über die maximale darstellbare Höhe von 100 hinausgeht, wird der Teiler verdoppelt und das Display mit dem neuen Teilerwert aktualisiert. Bei Jedem Impuls wird dann der Impulszähler erhöht und die Analogelektronik durch einen kurzen Impuls (Reset des Impuls-Flipflops) für weitere Impulse des SZR freigegeben.

```
void doSample(void){
    int temp;
    LED=1;
    temp=(doadc()/4);

    Incspec(temp);
    if(spec(temp)/Shalfts>100){
        Shalfts*=2;
        event=eDISPREN;
    }

    imps++;

    clear=0;
    clear=1;
    LED=0;
    return;
}
```

Abb. 12: C-Funktion "doSample()"

### 3.3 DA-Wandler

Die Abbildung 14 zeigt die Funktion „setSU()“, die die in „U“ angegebene Spannung an den DA-Wandler übermittelt. Dazu wird eine Clock- (DACclk), Chip-Select- (DACcs) und eine Datenleitung (DACdin) verwendet. Der DA-Wandler benötigt pro Spannungsänderung 24 Bit Daten, wobei 12 Bit für den Kanal A und die weiteren 12 Bit die Spannung für den Kanal B beinhalten.

In den ersten 10 Zeilen werden die Ports so gesetzt dass die Regeln, die im Datenblatt festgelegt sind, eingehalten werden. Gleiche, hintereinander folgende Zuweisungen sind dabei für das Timing entscheidend, da sie ihren Zustand so für die doppelte Zeit eines Taktzyklus halten.

In der nachfolgenden Schleife werden die ersten 12 Bit übertragen. Nachdem Clock auf 0 gesetzt wurde, wird das MSB (Most Significant Bit) zuerst gesendet, indem „temp“ mit „0x800“ auf das 12-te Bit maskiert und nach dessen Zustand der Datenpin gesetzt wird. Dann wird der Inhalt von „temp“ um ein Bit nach links verschoben, damit beim nächsten Schleifendurchlauf das nächste Bit maskiert wird. Danach wird die Clockleitung wieder auf

1 gesetzt und der Datenpin wird in das Shiftregister des DA-Wandlers übernommen. Nach insgesamt 12 Durchläufen werden die nächsten 12 Bit übermittelt, wobei der Datenpin permanent auf 0 steht, da der B-Kanal von mir nicht genutzt wird. Am Ende werden die Ports wieder in den „Ruhezustand“ gesetzt, damit der DA-Wandler die gesendeten Daten übernimmt und eine Spannung einstellt.

### 3.4 Display

Da das Display über eine Touchpadfunktion verfügt, verwende ich es nicht nur zur Ausgabe, sondern auch zur Eingabe von Daten. Es wird über die USART-Schnittstelle des PIC mit einem speziellen Protokoll (small protocoll) angesteuert und lässt sich so mit einfachen Befehlen steuern. Es kann auch über ein USB-Interface in einer „C“ ähnlichen Syntax separat programmiert werden. So lassen sich Bilder oder Fonts aufs Display übertragen. Auch lassen sich sogenannte Makros definieren, in denen sich zum Beispiel sich oft wiederholende Display-Formulare festlegen lassen, die später vom Mikrocontroller aufgerufen werden (z.B. ASCII-Tastatur zur Eingabe). Dies spart wertvollen Speicherplatz auf dem PIC.

Um den Programmablauf zu steuern, lassen sich Touch-Knöpfe zeichnen, die entweder mit Touchmakros verknüpft werden oder einfach einen bestimmten Tastencode senden.

Bevor die am Display betätigten Touchcodes allerdings an den PIC übermittelt werden, muss er erst eine Sendeaufforderung an das Display senden, erst dann wird der Sendepuffer des Displays geleert.

Diese Anfrage wird in regelmäßigen Abständen aus dem Timer-Interrupt erledigt.

```
void setSU(int U){
    int i;
    int temp;

    DACClk=1;
    DACCs=1;
    DACClk=0;
    DACCs=1;
    DACCs=0;
    DACCs=0;

    temp=U;
    for(i=0;i<12;i++){//DAC A
        DACClk = 0;
        if((temp & 0x800)==0)DACdin=0;
        else DACdin=1;
        temp = temp << 1;
        DACClk = 1;
        DACClk = 1;
    }

    temp=U;
    for(i=0;i<12;i++){//DAC B
        DACClk = 0;
        DACdin=0;
        DACClk = 1;
        DACClk = 1;
    }

    DACCs=1;
    DACCs=1;
    DACClk=0;
    DACCs=0;
    DACClk=1;
    return;
}
```

Abb. 13: Funktion "setSU()"

## 4 Praxis

### 4.1 Messungen

Abbildung 15 zeigt den Aufbau in der Schule, mit dem die folgenden Messungen durchgeführt wurden.

Das SZR und der Strahler wurden dabei mit Stativmaterial am Tisch befestigt, damit während der Messung definierte Bedingungen gegeben sind.

Hier werden die Messungen direkt auf dem Laptop gespeichert. Der graue Bleiblock dient zur Abschirmung der radioaktiven Strahlung.

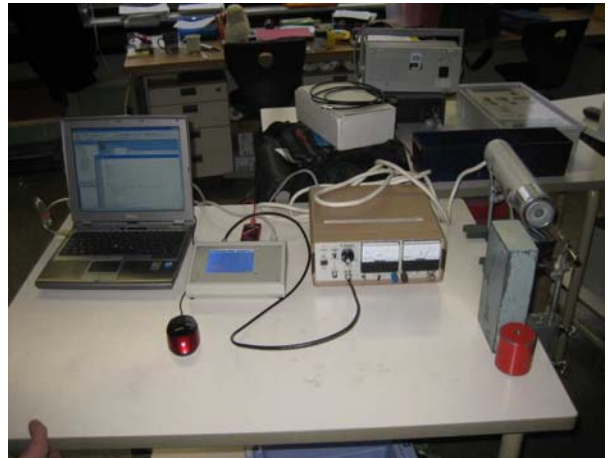


Abb. 14: Versuchsaufbau: von Links: Laptop, VKA, Hochspannungsnetzteil und SZR mit radioaktivem Präparat und Bleiblock zur Abschirmung

#### 4.1.1 Caesium 137

Das Diagramm (Abb. 16) zeigt ein Spektrum von Caesium 137 bei einer SZR-Spannung von 850V. Auf der X-Achse sind die Kanalnummern, auf der Y-Achse die Counts pro Kanal aufgetragen.

Im Bereich der Kanäle 0 bis 20 werden keine Ereignisse registriert, weil die Triggerung auf der Analogplatine so eingestellt ist, dass erst ab einer Mindestspannung Signale registriert werden, dies dient zur Unterdrückung des Zählrohrgrundrauschens. Zwischen Kanalnummer 20 und 50 lassen sich niederenergetische Effekte, wie sie bei der Wechselwirkung der Gammastrahlung mit der Materie entstehen, erkennen. Der Peak über Kanal 65 stammt vom Präparat. Die Aktivität des verwendeten Caesiums-Präparates liegt bei ca. 8000 kBq.

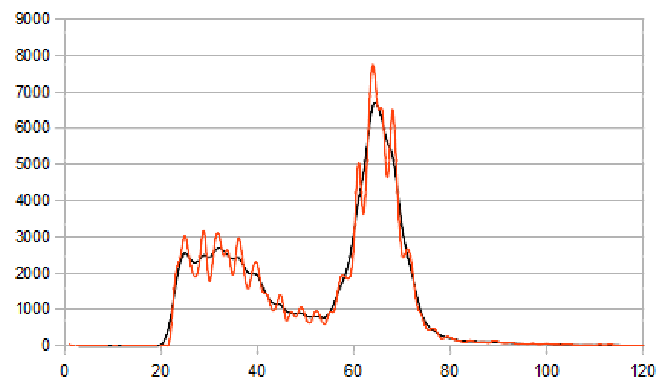


Abb. 15:  $^{137}\text{Cs}$ -Spektrum bei einer SZR-Spannung von 850 V. Orange: Original, Schwarz: Mittelwert

#### 4.1.2 Eichung

Da die Werte bei der Originalmessung ein starkes Rauschen haben, sollte man, um die Position des Peaks genau zu bestimmen, Mittelwerte der Messung bilden (Schwarze Linie in Abb. 16). Der Peak dieser Mittelwertkurve liegt bei Kanalnummer 67. Da die Zerfallsenergie von Caesium 137 (662keV) bekannt ist, kann man das Spektrum nun eichen. Die Kanalnummer zur Energie ist proportional, somit entspricht ein Kanal der Energie von Caesium durch die Peak-Position von Caesium:

$$662\text{keV} / 67 = 9,88\text{keV}$$

Jeder Kanal kann jetzt mit diesem Faktor multipliziert werden und man erhält die zugehörige Energie. Dieser Faktor ist für jede Zählrohrspannung anders und muss deshalb ebenfalls gespeichert werden.

### 4.1.3 Einfluss der SZR-Spannung auf das Spektrum

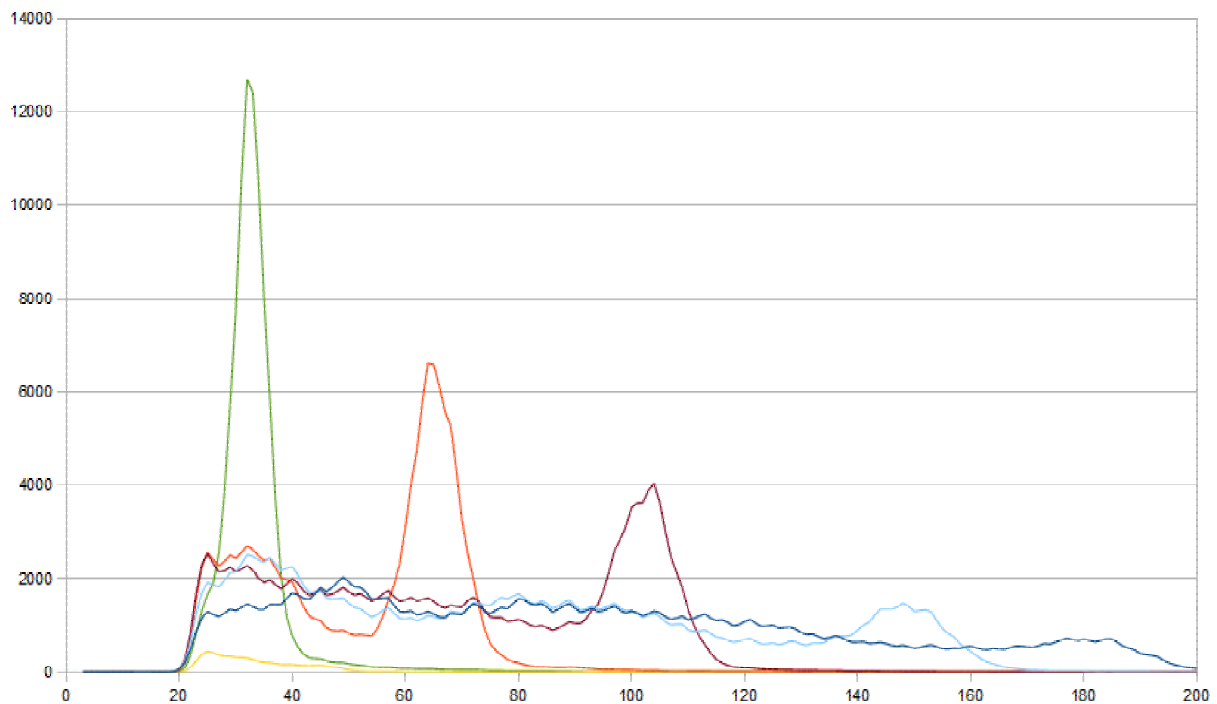


Abb. 16:  $^{137}\text{Cs}$ -Spektren (gemittelt) : Blau: 1000 V, Hellblau: 950 V, Violett: 900 V, Orange: 850 V, Grün: 800 V, Gelb: 750 V

In Abb. 17 sieht man sechs verschiedene Caesium-137 Spektren bei konstanten Messbedingungen, aber unterschiedlicher SZR-Spannung. Je höher die Spannung, desto größer ist der Abstand des Peaks von der Y-Achse und mehr Impulse mit niedrigerer Energie werden dargestellt. Deshalb gilt: Je höher die Spannung, desto höher die Empfindlichkeit der SZR. Bei 750V ist die Empfindlichkeit schon so gering, dass der Caesium-Peak kaum noch zu erkennen ist.

Mit steigender Spannung wird die Messung allerdings auch unschärfer und der Peak breiter, bis er bei 1000 V nicht mehr zu erkennen ist.

Damit man einerseits scharfe Peaks und andererseits auch eine hinreichende Empfindlichkeit erhält, sollte man mit diesem SZR zwischen 825 und 900 Volt messen. Zwischen 800 und 950 Volt ist der Zusammenhang zwischen Spannung und Abstand der Peaks zur Y-Achse nahezu linear.

### 4.1.4 Natrium 22

Das Isotop Natrium-22 ist unter anderem ein Beta+-Strahler. Wenn das bei dieser Strahlung emittierte Positron mit einem Elektron zusammenstößt, entstehen wegen der Paarvernichtung (Annihilation) 2 Gammaquanten mit einer Energie von genau 511 keV (Abb. 18). Natrium 22 kann deswegen ebenfalls gut als Eichpräparat verwendet werden.

Die Abbildung 19 zeigt das bei 950V aufgenommene Spektrum bei einer Messzeit von ca. 47 Minuten. Trotz

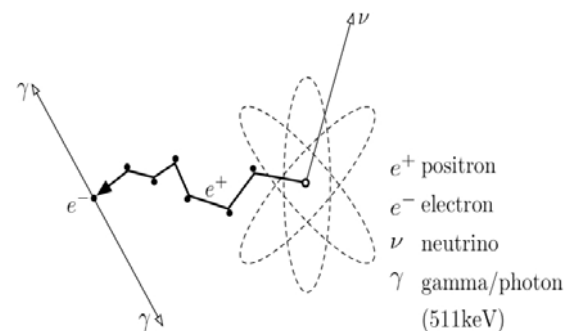


Abb. 17: Beispiel einer Annihilation (Gegenseitige Vernichtung eines Elektron und Positrons)

dieser relativ langen Messzeit wurden nur wenige Impulse gemessen, da die Aktivität des 25 Jahre alten Präparats aufgrund der geringen Halbwertszeit von 2,6 Jahren nur noch 0,24 kBq beträgt. Das ist 30.000 mal kleiner als die des verwendeten Caesiumpräparates. Der Peak zwischen 360 und 420 keV ist auf den Compton-Effekt zurückzuführen, bei dem Gammaquanten durch Stöße mit Elektronen einen Teil ihrer Energie abgeben und die gestreuten Quanten vom SZR registriert werden.

Am Ende der Messung wurde das Ergebnis zunächst gespeichert (Blau) und anschließend das Cs137-Präparat einige Sekunden vor das SZR gehalten, wobei die Messung fortgesetzt wurde. Diese Methode ermöglicht eine Eichung (Orange) der Energieskala. Auf der mit dem Caesium geeichten X-Achse liegt der Natrium-Peak exakt bei 511 keV.

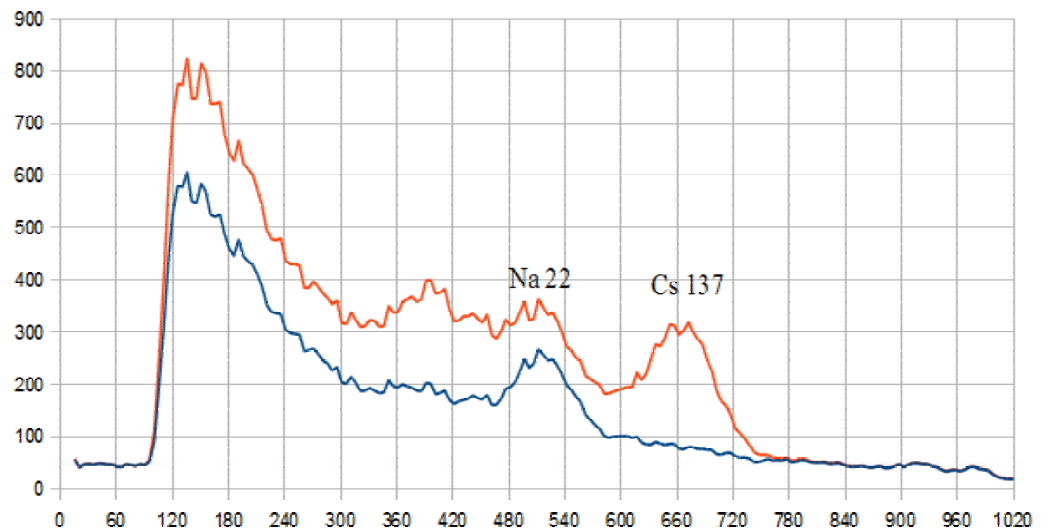


Abb. 18: Na22-Spektrum: bei 950 V, 47 min, Orange: mit kurzem Caesium-Einfluss, Y-Achse: Counts, X-Achse: Energie in keV

#### 4.1.5 Cobalt 60

Das Isotop Cobalt-60 zerfällt mit 2 verschiedenen Gammaenergien, die bei 1170 und 1330 keV liegen, wobei die Häufigkeit der Impulse bei 1330 keV geringer ist.

Die beiden Peaks der entsprechenden Energien lassen sich eindeutig erkennen. Der etwas breitere Peak um 700 keV lässt sich wieder (vergl. 4.1.4) auf den Compton-Effekt zurückzuführen.

Die Aktivität des 41 Jahre alten Präparats beträgt bei einer Halbwertszeit von 35 Jahren ca. 657 kBq.

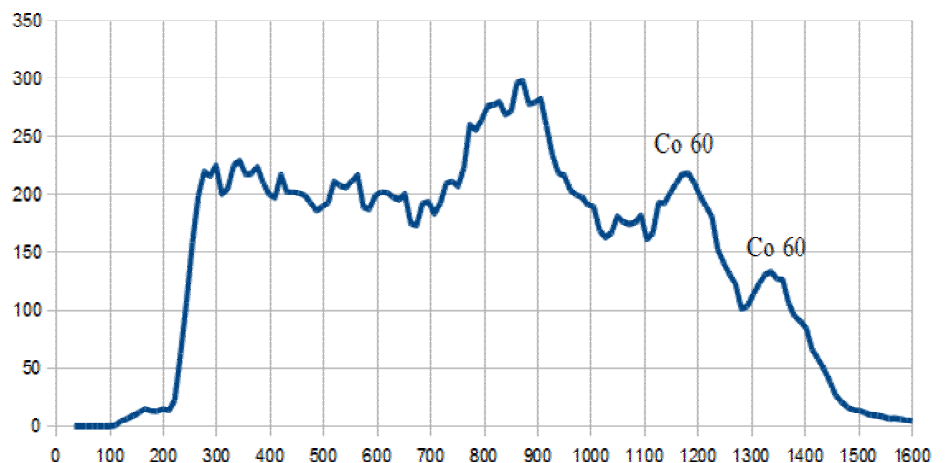


Abb. 19: Co60-Spektrum: Bei 850 V, ca. 16 min, Y-Achse: Counts, X-Achse: Energie in keV

### 4.1.6 Umgebungsstrahlung

Die Abbildung 21 zeigt eine Messung bei 1000 V ohne ein radioaktives Präparat in der Nähe. Die gemessenen Impulse stammen daher von Umgebungsstrahlung. Diese Strahlung kann zum Beispiel von natürlichem radioaktiven Radon stammen oder kosmische Strahlung aus Sonnenwinden sein.

Der kleine Peak bei 715 keV ist auf den in 2.2 beschriebenen Messfehler, der durch die zu frühe Triggerung hoher Impulse entsteht, zurückzuführen, der bei den geringen Zählraten besonders auffällt.

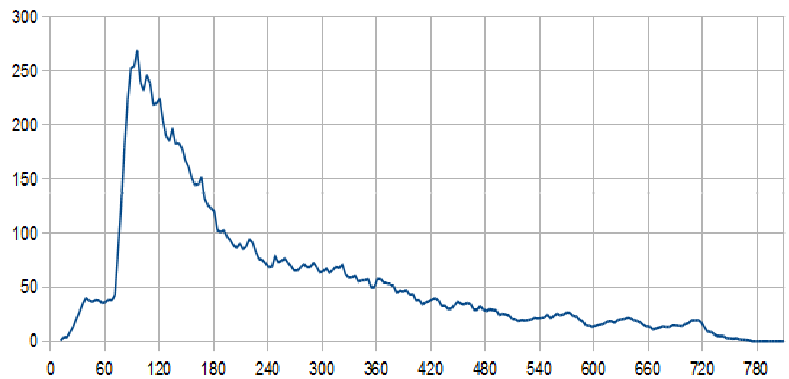


Abb. 20: Messung ohne Präparat bei 1000 V, 30 min  
Y-Achse: Counts, X-Achse: Energie in keV

## 4.2 Kosten

Mein Ziel war es auch, das Gerät möglichst günstig zu entwerfen. Während sich handelsübliche VKAs mit ähnlichem Funktionsumfang im Preisbereich von 1200€ bewegen, kostet mein VKA mit Elektronik, Display (120 €) usw. ca. 200 €. Diese Summe unterschreitet deutlich die der käuflichen VKAs.

## 5 Quellen, Literatur, Werkzeuge

- Datenblätter der Firma Microchip:
  - MPLAP\_C18\_Libraries.pdf (Handbuch zum MCC18 C-Compiler)
  - PIC18Fxx2 datasheet.pdf (Datenblatt zum PIC18F452)
- Software der Firma Microchip:
  - MPLAB IDE v8.36 (Entwicklungsumgebung für Microchip PICs)
  - MCC18 v3.34 (C-Compiler für Microchip PICs)
- Schaltpläne erstellt mit
  - Eagle 4.16 r2 Light (Frei verfügbare Version)
- Genaue Werte der Gammaenergien
  - de.Wikipedia.org/wiki/Natrium
  - de.Wikipedia.org/wiki/Cobalt
  - de.Wikipedia.org/wiki/Caesium
- Sonstige Informationen, Bilder
  - <http://www.biedermann-weesen.de/Skripte/Kernphysik.pdf>

## 6 Danksagung

Ich bedanke mich besonders bei meinem Betreuungslehrer Thomas Biedermann, für seinen großen Zeiteinsatz und seine Geduld. Auch bedanke ich mich natürlich bei Susanne Biedermann für die tolle Verpflegung von morgens bis abends. DANKE!